Neuro-Symbolic Artificial Intelligence: A Reference Model for the Design and Development of Neuro-Symbolic Systems

BRANDON C. COLELOUGH*, Department of Computer Science, University of Maryland, College Park, USA

WILLIAM REGLI, Department of Computer Science, University of Maryland, College Park, USA

Neuro-Symbolic Artificial Intelligence (NSAI) lacks a grounded reference model that can guide design across a range of systems that differ significantly in task settings, operating environments, and design requirements. We derive a descriptive, engineering-oriented reference model for functional Neuro-Symbolic systems from independently rebuilt and rerun Neuro-Symbolic systems. We define a functional Neuro-Symbolic system as one whose released materials allow the reconstruction of the environment and workflow, whose codebase executes as a complete workflow, and whose rerun still supports the main published claim. The PRISMA-guided review analyzed 1,304 records and yielded a verified corpus of 85 functional systems. Recurring structure across that corpus resolved to an emergent architecture centered on a Neuro-Symbolic Environment within a Neuro-Symbolic Problem-Set Domain that employs a neural component, an explicit symbolic store, a reasoning process, and a controller. Our emergent Neuro-Symbolic Reference Model is organized into an ordinal Level 0–7 scale and six role-preserving architectural families. Foundational to our proposed Neuro-Symbolic Reference Model is the claim that the build-order of any Neuro-Symbolic system is critical to the successful development and overall usefulness of that Neuro-Symbolic system once built. Our proposed reference model ensures Neuro-Symbolic systems are comparable by roles, boundaries, information flow, and build dependencies.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; *Knowledge representation and reasoning*; *Neural networks*; • **Software and its engineering** → General programming languages.

Additional Key Words and Phrases: Neuro-Symbolic Artificial Intelligence, Neuro-Symbolic Systems, Reference Model, Functional Systems, System Architecture, System Design, Systematic Literature Review, Reproducibility

ACM Reference Format:
Brandon C. Colelough and William Regli. 2026. Neuro-Symbolic Artificial Intelligence: A Reference Model for the Design and Development of Neuro-Symbolic Systems. *ACM Comput. Surv.* 1, 1 (June 2026), 49 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Although there has been increasing interest in the field of Neuro-Symbolic AI, and recent review work reports rapid growth in publications since 2020 and a growing body of surveys that attempt to organize the area [20], the broad and expanding domain of the NSAI literature currently does not have a grounded reference model that informs intelligent design decisions for Neuro-Symbolic systems. Practitioners apply the Neuro-Symbolic label to systems that differ substantially in task

*Brandon Colelough is the corresponding and first author. He led the systematic literature review, the reproducibility study, and the framework to define the Neuro-Symbolic Reference Model.

Authors' Contact Information: Brandon C. Colelough, brandcol@umd.edu, Department of Computer Science, University of Maryland, College Park, College Park, Maryland, USA; William Regli, regli@umd.edu, Department of Computer Science, University of Maryland, College Park, College Park, Maryland, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1557-7341/2026/6-ART
https://doi.org/XXXXXXX.XXXXXXX

setting, operating environment, and roles, which makes it difficult to determine which design choices are actually necessary for a Neuro-Symbolic system to function. Existing survey and taxonomy efforts describe themes, trends, and application areas, but they do not establish a common architectural account grounded in systems that independent reruns have verified as working systems. Without a grounding reference model for Neuro-Symbolic systems, we as a community cannot clearly establish and delineate real, functional Neuro-Symbolic systems.

1.1 Purpose

The paper derives a reference model for functional Neuro-Symbolic systems from independently rebuilt and rerun Neuro-Symbolic systems. The aim is descriptive and engineering-oriented. The reference model description abstracts recurring structures from verified systems and uses those structures to state a common architecture, an ordinal-level scheme, recurrent architectural families, and a build-order argument grounded in worked cases.

1.2 Scope

We define a functional Neuro-Symbolic system as one whose released materials allowed reconstruction of the environment and workflow, whose code ran end-to-end, and whose rerun still supported the original manuscript's main published claim.

1.3 Contributions

The paper contributes four main artifacts.

(1) A common Neuro-Symbolic Reference Model derived from recurrent structure across the verified corpus.
(2) An ordinal Level 0 through Level 7 scheme derived from the shared base model of the Neuro-Symbolic Reference Model.
(3) Six role-preserving architectural families of functional Neuro-Symbolic systems that make systems comparable by dominant coupling pattern without reducing them to one implementation template.
(4) A build-order argument for Neuro-Symbolic system development.

1.4 Definitions

Definition 1.1 (Neuro-Symbolic system). A Neuro-Symbolic system is an AI system in which a controller coordinates functions between a neural component, an explicit symbolic store, and a reasoning process within an operating environment to generate valid solutions for a given Problem-Set Domain.

The Problem-Set Domain fixes the class of tasks, the symbolic objects, the constraints, and the success conditions that make an output valid. The operating environment provides the world in which observations, states, actions, feedback, and consequences are available. Within that setting, the neural component forms representations and proposes candidate answers, steps, programs, proofs, or actions, the symbolic store maintains explicit knowledge and structured state, and the reasoning process derives, checks, plans, proves, executes, or repairs across training and runtime. The system boundary distinguishes the input and output of the internal system from the broader external flow of input data and output information, and an external orchestrator may mediate that flow without becoming part of the internal control loop.

1.5 Manuscript Structure

This paper is organized as follows. Section 2 summarizes the verified Neuro-Symbolic AI literature from 2019 through 2025. Section 3 describes the PRISMA review, reproducibility assessment, and synthesis procedure used to derive the model. Section 4.2 presents a high-level overview of the 85 verified systems. Section 5 presents the reference model, including component roles, data interfaces, the ordinal level scheme, and the build-order argument. Section 6 describes six architectural families of functional Neuro-Symbolic systems. Section 7 discusses practical design implications and worked examples. Section 8 concludes.

2 Background

2.1 A timeline of the verified Neuro-Symbolic systems

The verified Neuro-Symbolic systems literature develops in identifiable stages within the broader post-2020 expansion of Neuro-Symbolic Artificial Intelligence, documented by recent review work [20]. The timeline described below does not follow nor account for every self-described NSAI paper, but instead describes the timeline for only the 85 systems retained in the verified corpus used in this paper. The verified literature timeline shows not a single line of progress toward one canonical architecture, but a cumulative expansion from early symbolic reasoning substrates and latent symbolic induction toward richer environments, stronger verification regimes, and more explicit controller logic. The earliest verified systems established the base pattern that later working systems preserved. Neural Logic Machines and the visual discrimination system already showed, by 2019 (later revised and republished in 2022), that neural processing could feed an explicit symbolic representation on which the main task-solving logic operated [27, 68]. The first 2020 wave then pushed that pattern into generative concept modeling, closed-loop correction, planning-model induction, and ontology-grounded semantics through Learning Task-General Representations with Generative Neuro-Symbolic Modeling, Closed loop Neuro-Symbolic learning, Learning Neuro-Symbolic descriptive planning models via cube-space priors, and Semantic similarity and machine learning with ontologies [8, 33, 53, 57]. The 2021 cohort then widened the verified corpus into theory induction, rule induction, symbolic control, scientific regression, and structured data construction through Making sense of raw input, Neuro-Symbolic Hierarchical Rule Induction, Tunable Neural Encoding of a Symbolic Robotic Manipulation Algorithm, Neural symbolic regression that scales, and Knowledge-driven Data Construction for Zero-shot Evaluation in Commonsense Question Answering [14, 31, 35, 45, 62]. The historical importance of this formative period is that it already fixed the recurrent division of labor that survives across the rest of the verified timeline. Neural components propose or parameterize candidate structures, while symbolic structures supply the objects, constraints, and executable checks that make those candidates usable. The 2022 verified literature widened Neuro-Symbolic system design from early induction pipelines into probabilistic reasoning, differentiable constraints, structured language modeling, and runtime logic graphs. Logic Tensor Networks, DeepStochLog, A-NeSI, Neuro-Symbolic entropy regularization, Semantic probabilistic layers for Neuro-Symbolic learning, Embed2Sym, Knowledge Enhanced Neural Networks for relational domains, Neuro-Symbolic language modeling with automaton-augmented retrieval, Neuro-Symbolic Natural Logic with Introspective Revision for Natural Language Inference, ConvFinQA, VAEL, and AdaLoGN all belong to this phase [3, 4, 6, 9, 10, 18, 23, 34, 58, 67, 82, 83]. The 2022 cohort of real functional Neuro-Symbolic systems made symbolic structure do more than post hoc checking. In these systems, logic and constraints enter the computational graph, shape learning, structure inference, or generate intermediate symbolic extensions inside the runtime loop itself. The 2023 cohort diversified the verified literature across event processing, solver-backed reasoning, symbolic programming languages, ASP pipelines, structured story state, image program synthesis, motion programs, commonsense reasoning, and natural-language automated reasoning. DeepProbCEP, Interpretable Neuro-Symbolic Concept Reasoning, A multi-grained self-interpretable symbolic-neural model for single or multi-labeled text classification, Scalable Neural-Probabilistic Answer Set Programming, Learning to Solve Constraint Satisfaction Problems with Recurrent Transformer, NeuPSL, Injecting Logical Constraints into Neural Networks via Straight-Through Estimators, Dynamic Planning with an LLM, Learning Signal Temporal Logic through Neural Network for Interpretable Classification, There and Back Again, A Modular Neuro-Symbolic Approach for Visual Graph Question Answering, Scallop, Empirical Investigation of Neural Symbolic Reasoning Strategies, Leveraging large language models to generate answer set programs, LINC, CORRPUS, ImageEye, Motion Question Answering via Modular Motion Programs, Neuro-Symbolic Commonsense Social Reasoning, and An Experimental Pipeline for Automated Reasoning in Natural Language all appear in this stage of the verified record [7, 12, 13, 16, 21, 28–30, 39, 41, 46, 56, 60, 70, 74, 76–78, 86, 87]. By 2023, working Neuro-Symbolic systems already span pipeline reasoners, constraint-shaped learners, joint inference solvers, controller-led planning loops, and artifact synthesis systems, all within the same verified corpus. The 2024 cohort is the broadest and most differentiated stage in the verified timeline. One part of this cohort deepened controller logic, verification, and runtime orchestration through Error Detection and Constraint Recovery in Hierarchical Multi-Label Classification without Prior Knowledge, LogiCity, The Constitutional Filter, Think before You Simulate, Mitigating data sparsity via Neuro-Symbolic knowledge transfer, CodePlan, NeSy is alive and well, Neuro-Symbolic Embedding for Short and Effective Feature Selection via Autoregressive Generation, A Walsh Hadamard Derived Linear Vector Symbolic Architecture, Neuro-Symbolic Repair of Test Flakiness, Convex and Bilevel Optimization for Neuro-Symbolic Inference and Learning, SymbolNet, and Vehicle [1, 5, 11, 15, 17, 22, 26, 36, 42, 48, 51, 55, 80]. The 2024 cohort also pushed the verified literature toward formal reasoning, knowledge-rich biological and relational systems, symbolic mathematics, and repository or query level orchestration through NeuroComparatives, KLay, Protein function prediction as approximate semantic entailment, Formal verification of parameterised Neuro-Symbolic multi-agent systems, Relational programming with foundation models, MARS, DiaKoP, Autoformalizing euclidean geometry, LARS-VSA, Accelerating UMR adoption, Neuro-Symbolic Training for Reasoning over Spatial Language, LLMMeetsBoundedModelChecking, Enhancing SQL Query Generation with Neuro-Symbolic Reasoning, MILE, and AlphaGeometry (Solving olympiad geometry without human demonstrations) [24, 38, 50, 52, 54, 59, 63, 66, 69, 71–73, 79, 84, 85]. The same year also includes BEARS, which makes shortcut awareness part of the verified literature even though its symbolic contribution remains lighter than the stronger reasoning systems in the corpus [65]. 2024 marks the point at which the verified literature most clearly connects Neuro-Symbolic design to hard symbolic validation, repository-scale software change, executable test or proof environments, and more explicit controller responsibilities. The 2025 cohort extends the verified Neuro-Symbolic literature domains into more niche, problem-specific areas such as diffusion models, mission design, automata, travel demand prediction, digital twins, theorem-proving refinement, LLM rule guidance, scientific discovery, association rule mining, repository translation, neuro-conceptual question answering, and dataset or agent construction. Neuro-Symbolic Diffusion Models, Probabilistic Mission Design for Neuro-Symbolic Unmanned Aircraft Systems, NeSyA, Neuro-Symbolic AI for Travel Demand Prediction, ANSR-DT, Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning, Improving Rule-based Reasoning in LLMs using Neuro-Symbolic Representations, Neural Symbolic Model for Space Physics, Neuro-Symbolic Association Rule Mining from Tabular Data, PEIRCE, AlphaTrans, Neuro-Conceptual Artificial Intelligence, and MDD-5k occupy that most recent verified layer [2, 25, 37, 40, 43, 44, 49, 61, 64, 75, 81, 88, 89]. Across the full verified timeline, the literature broadens substantially in task domain, environment design, reasoning depth,

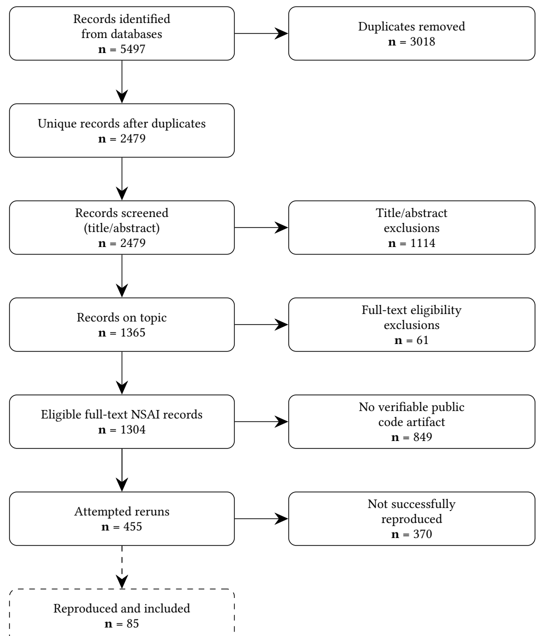

Fig. 1. Selection and eligibility flow for the Neuro-Symbolic audit. Late full-text eligibility exclusions are shown separately and are not counted as rerun outcomes. The dashed bottom box reports the number of successful reruns within the attempted set and is shown as an audit outcome.

and controller strength. The verified corpus of 85 functional Neuro-Symbolic systems accumulates many working realizations of the same underlying pattern, Neuro-Symbolic sub-component roles fixed within a Neuro-Symbolic environment supported by a Neuro-Symbolic Problem-Set Domain, motivating our research for deriving a grounded reference model from the verified corpus itself.

3 Methods

3.1 Scoping review and vocabulary development

A 2024 PRISMA-guided systematic scoping review [20]mapped the post-2020 Neuro-Symbolic AI literature and established the initial search vocabulary and thematic baseline used in subsequent stages. The review searched five sources (IEEE Xplore, Google Scholar, arXiv, ACM Digital Library, and SpringerLink) covering publications from 2020 through 2024. From 1,428 initial records, duplicate removal and title-and-abstract screening retained 392 candidates, of which 167 had publicly available code repositories. Full-paper screening removed a further 9, leaving 158 papers in the scoping corpus. The review organized the post-2020 Neuro-Symbolic literature around five thematic areas, including knowledge representation, learning and inference, explainability and trustworthiness, logic and reasoning, and meta-cognition. These thematic areas and the search vocabulary developed from them informed the terminology used in the subsequent updated systematic review, but the scoping corpus supplied no functional verification of any system and did not produce the architectural extraction codebook used to derive the reference model.

3.2 Updated corpus construction and eligibility screening

The second evidence stage was a protocol-guided updated systematic literature review followed by a same-artifact reproducibility audit, searching nine digital libraries and preprint repositories through 23 May 2025 [19]. The nine sources were Web of Science, Scopus, PubMed, EI Compendex, IEEE Xplore, ACM Digital Library, SpringerLink, Google Scholar, and arXiv, queried with broad terms covering neuro-symbolic, neurosymbolic, neural-symbolic, symbolic-neural, Neuro-Symbolic, and NeSy. Searching nine sources (increased from the five used in the scoping review) broadened literature coverage of emerging and lexically idiosyncratic work that narrower queries and fewer repositories would have missed, at the cost of substantially higher deduplication and title-and-abstract screening workload burden. The wider source set was necessary because the reference model required candidates drawn from a codebase-verified corpus of Neuro-Symbolic systems beyond the 2024 scoping review. Screening separated unique, on-topic, and full-text eligible Neuro-Symbolic records through a sequence of documented reduction steps, as shown in Figure 1. Deduplication reduced 5,497 retrieved records to 2,479 unique records, and title-and-abstract screening against Neuro-Symbolic relevance criteria retained 1,365 on-topic records. Full-text eligibility assessment then removed 61 records for reasons including off-topic framing, absence of quantitative evaluation, and inaccessible full text, leaving 1,304 eligible full-text Neuro-Symbolic records. The 61 late full-text eligibility exclusions are analytically distinct from rerun outcomes and are not counted as audit failures. A paper that passed full-text eligibility but failed to reproduce under the same-artifact protocol entered the attempted rerun pool and received an audit outcome label, not a retroactive eligibility exclusion. The public-artifact gate was a construct validity criterion because a reference model for functional Neuro-Symbolic systems cannot be grounded in systems whose operation can only be described in print. Repository verification required each of the 1,304 eligible records to provide, or be matchable to, a verifiable public code artifact for the reported system. Records without a verifiable public artifact cannot support independent inspection, reconstruction, or execution of the reported Neuro-Symbolic workflow, which means the architectural roles and coupling patterns they describe remain unverified claims. Applying the public-artifact gate removed 849 records and left 455 code-bearing studies eligible for the same-artifact rerun, establishing the pool from which verified functional systems could be drawn.

3.3 Same-artifact reproducibility audit

Each of the 455 code-bearing studies entered a same-artifact reproducibility audit that tested whether an independent team could recover the reported claim using only the artifacts the authors released [19]. The audit followed a staged protocol covering repository and artifact verification, environment reconstruction, executable-integrity testing, result re-execution, post-rerun data extraction, author-contact procedures for inaccessible essential artifacts, quality-control review, and log archival [19]. Title and abstract screening was double-blind, with each record independently coded by two annotators and disagreements resolved by discussion. A ten-paper calibration pilot produced a Cohen's $\kappa = 0.82$ before full auditing began. Same-artifact reruns were single-annotator, with workload randomly allocated across the eight-member team. Edge cases and borderline repair decisions were reviewed in biweekly reconciliation meetings. Audit outcomes were recorded as post-entry labels, keeping the screening chain and rerun results analytically distinct.

3.3.1 Artifact requirements. The audit treated a public code link as insufficient unless the released materials formed a complete and connected runnable workflow for the primary experiment [19]. A qualifying artifact set had to supply source code, data or fixed splits, model weights or checkpoints when the evaluation depended on a fixed trained model state, environment specifications, evaluation scripts or assets, preprocessing scripts that alter data semantics, rule sets or knowledge bases when those were indispensable to the primary experiment, configuration files, and documentation sufficient to connect all of the foregoing materials into an executable pipeline. An artifact is counted as missing only when the artifact was indispensable to rerunning the primary experiment and could not be reconstructed from the paper and the released materials under the protocol. That standard excluded nominal code releases that described a system without supplying the materials an independent team would need to execute it, and it concentrated the audit on the gap between public availability and actual "runnability". Papers that passed this artifact check entered the repair-boundary and outcome-label stages of the protocol.

3.3.2 Permitted and prohibited repairs. Minimal maintenance repairs were allowed when the repairs preserved the original Neuro-Symbolic architecture and workflow without altering the model or experimental object [19]. Permitted repairs covered four categories, including updating deprecated package dependencies to versions that restored the intended build, correcting file path references that had broken due to repository reorganization, supplying a lightweight evaluation harness when the intended evaluation workflow was unambiguous from the paper and code, and applying minor scripting patches that corrected syntactic errors without touching model logic. Repairs that passed the maintenance threshold left the Neuro-Symbolic architecture, the training procedure, the dataset, and the experimental claim intact, so the rerun remained attributable to the authors' design. Repairs that crossed the model boundary invalidated the same-artifact rerun and produced a non-reproducible audit outcome. Prohibited changes included altering the Neuro-Symbolic architecture in any way, correcting bugs intrinsic to the model architecture, substituting or reconstructing missing indispensable artifacts that the authors had not released, inventing or augmenting data, and reinterpreting the paper's experimental claim to match what the released code actually computed.

3.3.3 Outcome labels and success threshold. The audit assigned outcome labels as post-entry rerun results, keeping the screening chain and the rerun record analytically distinct [19]. Each study that entered the attempted rerun pool received one of five labels after the primary experiment ran or failed to run. O1 marked a fully reproduced study, O2 marked a partially reproduced study, O3 marked a study that executed but did not reproduce within the accepted tolerance, O4 marked a study that could not execute because one or more indispensable artifacts were missing or inaccessible, and O5 marked a study that could not execute because the released code or environment contained irrecoverable defects. All five labels are audit outcomes, not eligibility exclusions, and a study assigned O3, O4, or O5 remained part of the documented attempted rerun pool. Reproduction success for model construction required an outcome of O1 or O2, and the two labels differed in what the rerun had to preserve. O1 required the primary experiment to execute successfully and to reproduce the reported primary result within the fidelity criterion, defined as agreement within plus or minus 5 percent absolute error or within the authors' reported 95 percent confidence interval, without correcting any model-architecture bug. The fidelity criterion and its justification are documented in full in the companion reproducibility study [19]. O2 required the core pipeline to execute and the paper's main qualitative claim to remain supported, even when some quantitative results or secondary experiments could not be matched exactly. Both O1 and O2 reruns ran with the author-specified compute configurations wherever the released documentation specified them. Studies assigned O3, O4, or O5 did not contribute to the model-building corpus because none of the three outcomes preserved executable support for the authors' published claim. An O3 study was run, but produced results that contradicted or materially exceeded the tolerance of the reported claim, meaning the released workflow did not reproduce what the paper claimed. O4 and O5 studies could not be executed at all, so no architectural evidence about the running system could be extracted.

3.4 Functional-system inclusion criterion

A study entered the reference-model corpus only when the released artifacts supported reconstruction of the computational environment, execution of the workflow from input to output without missing steps, and preservation of the main published claim under independent rerun [19]. Preservation of the main published claim meant that the rerun produced outputs consistent with what the paper reported as the primary result, within the accepted fidelity criterion, without correcting any defect intrinsic to the Neuro-Symbolic architecture itself. A paper could be topically relevant to Neuro-Symbolic AI and still fail the functional-system gate if the released materials could not be connected into a runnable workflow or if the rerun contradicted the published claim. The gate, therefore, measures executable claim support. Applying the functional-system gate to 455 attempted reruns yielded 85 verified functional systems, as shown in Figure 1. Of the 85 retained systems, 48 received outcome label O1 as fully reproduced systems and 37 received label O2 as partially reproduced systems. The 370 studies that did not pass the gate received outcome labels O3, O4, or O5 and contributed no architectural evidence to the reference model. The 61 records that failed late full-text eligibility checks are counted separately in the eligibility funnel and are not part of the attempted rerun pool. Together, the 85 verified functional systems represent 18.68 percent of the 455 attempted reruns and 6.52 percent of the 1,304 eligible full-text Neuro-Symbolic records. The functional-system gate separates claim-level Neuro-Symbolic relevance from the verified functional inclusion required to ground a reference model. A paper could describe a Neuro-Symbolic architecture, appear in a peer-reviewed venue, and include a public code link, yet still fail the gate because the released workflow could not be reproduced or because the rerun did not reproduce the published claim. Conversely, lower-boundary systems remained eligible for the corpus when the reproduced workflow contained an active neural-symbolic coupling and the rerun supported the original claim, even when the Neuro-Symbolic contribution was lighter than in stronger systems. Retaining lower-boundary systems under that condition preserved the full observed range of integration depth in the verified-system corpus, which is covered by the ordinal-level scheme and architectural families.

3.5 Extraction codebook and architectural decomposition

Each of the 85 reproduced systems was converted into a normalized architectural record before any cross-system abstraction began, separating implementation-specific terminology from the role-level structure the reference model must capture. Converting a reproduced system into a normalized record meant replacing the paper's own vocabulary (which varied across venues, task domains, and research communities) with a fixed set of architectural fields. Without that normalization step, comparing a theorem-proving system against a structured-prediction system or a visual-reasoning system would conflate surface naming differences with genuine architectural differences, producing an abstraction that reflected vocabulary variation. The normalized record for each reproduced system, therefore, became the unit of evidence on which the cross-system abstraction operated.

3.5.1 Normalized extraction fields. The extraction record made each reproduced system comparable by role-level structure. Normalizing across papers required a fixed field set that mapped any implementation vocabulary onto the architectural positions defined in Definition 1.1, so that a system described using logic programming terminology and a system described using probabilistic inference terminology could be placed in the same record format without collapsing genuine architectural differences into a naming artifact. Table 1 maps each field to its specific purpose in reference-model derivation, and the field set was refined iteratively against the full verified-system corpus until every retained system could be represented without adding paper-specific slots. A field that could not be filled for a given system received a role-status code, preserving the record's

Table 1. Normalized extraction fields used to derive the Neuro-Symbolic reference model.

Extraction field	**Purpose in model derivation**
Problem-set domain	Identifies the class of tasks, symbolic objects, constraints, and success conditions that make an output valid.
Operating environment	Identifies the operative world in which the system receives inputs, maintains state, observes feedback, acts, and produces consequences.
Neural / sub-symbolic component	Identifies the learned component that forms representations or proposes candidate answers, steps, programs, proofs, actions, or constructions.
Symbolic store	Identifies the explicit knowledge or structured state used by the system, including facts, rules, constraints, programs, schemas, graphs, proofs, or ontologies.
Reasoning process	Identifies the process that derives, checks, plans, proves, executes, verifies, or repairs candidate results.
Controller/orchestration	Identifies how the system coordinates the neural component, symbolic store, reasoning process, and release of outputs across training and/or runtime.
System input and output	Identifies the internal system boundary: what enters the Neuro-Symbolic loop as system input and what leaves it as validated system output.
Verification/validation mechanism	Identifies how outputs or intermediate candidates are checked, including benchmark evaluation, formal checking, constraint checking, execution-based validation, or consistency testing.
Primary Neuro-Symbolic level	Records the system's dominant position in the Level 0–7 ordinal scheme based on integration strength, reasoning depth, controller involvement, and verification structure.
Architectural family	Records the system's dominant role-preserving coupling pattern so that systems can be compared without reducing them to one implementation template.

structural completeness even for lower-boundary systems. The extraction fields connect system operation to the structural positions that the reference model must account for. Each normalized record captured the Problem-Set Domain, the operating environment, the controller, the neural or sub-symbolic component, the symbolic store, the reasoning process, the system input, the system output, the verification or validation mechanism, the architectural family, and the primary Neuro-Symbolic level. The first two fields locate the system in the outer layers of the reference model, where task class, symbolic success conditions, and operative world must be specified before the inner components can function. The next six fields decompose the internal Neuro-Symbolic loop into the roles that Definition 1.1 requires. The final three fields position each system within the ordinal level scheme and the six architectural families, connecting the per-system record to the cross-system abstraction that produces the reference model.

3.5.2 Role-status coding. Role-status coding distinguished active, no-op, and absent roles so that lower-boundary systems could be represented in the shared extraction record without weakening the functional-system gate. A role received the status active when the reproduced workflow showed the role materially transforming, storing, deriving, verifying, repairing, or coordinating system state during the primary experiment. An active status required evidence from the reproduced workflow, released artifacts, and architectural record that the role materially transformed, stored, derived, verified, repaired, or coordinated system state. When evidence was insufficient to distinguish no-op from absent, the assignment was marked unclear and resolved conservatively during reconciliation.

No-op coding preserved the structural presence of a role in the extraction record without crediting the role with functional contribution. A role received the status no-op when the reproduced workflow contained an explicit implementation for the role with a defined input-output interface, the role was reachable during execution of the primary experiment, and the role returned an identity, pass-through, empty, or fixed result without materially transforming system state in the evaluated configuration. The absent status applies when no observable implementation, interface, or reachable workflow position for the role could be identified in the released artifacts. A no-op role confirms that the architectural position exists and was wired into the system, while an absent role means the system does not instantiate that position at all. No-op coding alone could not substitute for a missing active neural-symbolic coupling, and systems remained eligible for the verified-system corpus only when the reproduced workflow contained at least one active neural-symbolic coupling and still supported the paper's main published claim. A system in which the neural component, symbolic store, and reasoning process all received no-op status would fail to meet Definition 1.1 because the internal loop would produce no substantive Neuro-Symbolic operation, leaving the controller with no active components to coordinate. Role-status coding, therefore, served to represent structural variation across the corpus without relaxing the minimum functional conditions that were present and observed in the verified-system corpus.

3.6 Cross-system abstraction and synthesis

The synthesis procedure followed reference-model construction logic by first representing each of the 85 verified functional systems in the shared extraction vocabulary and only then abstracting recurring roles, information flows, and dependencies into a model. Representing concrete systems in a common vocabulary before abstracting is the method that separates technology-independent structural claims from implementation-specific observations, because abstraction applied directly to paper text confounds terminology variation with architectural variation. The synthesis proceeded in three reconciliation passes. In the first pass, each of the 85 normalized records was reviewed independently, and any system that could not be represented without adding a paper-specific slot triggered a vocabulary revision. Proposed fields were checked against the full verified corpus before being retained, and a field was kept only if it captured a role needed across multiple systems. In the second pass, all records were re-examined against the revised vocabulary to resolve outstanding role-status conflicts. In the third pass, coupling patterns, integration depth, and verification structures were compared across records to derive the level assignments and family groupings. Where ambiguity remained, lower Neuro-Symbolic levels were preferred, and conflicts were resolved conservatively. The vocabulary was considered stable when a complete pass through all 85 records produced no new role fields and no unresolved conflicts. Table 2 maps each stage of the evidence pipeline (from the 2024 scoping review through cross-system synthesis) to the specific model output each stage produced. The constraint on the abstraction procedure was that every retained system had to be representable in the common role vocabulary without adding paper-specific slots, which meant the vocabulary had to expand to cover genuine architectural variation. Roles that were required to represent every verified system in the shared vocabulary became reference-model components. Recurrent differences in how those roles coupled (whether the neural component fed a downstream symbolic engine, shaped learning through a differentiable objective, or operated inside an iterative repair loop) became the six architectural families. Differences across systems in integration strength, feedback depth, controller involvement, and verification regime formed a natural ordering that became the Level 0 through Level 7 ordinal scheme. Observed dependencies among the Problem-Set Domain, operating environment, core components, controller, and verification mechanism, where the outer layers had to be specified before the inner components could function coherently, became the build-order argument grounded in the verified corpus. We derived the

Table 2. Corpus-to-model derivation pipeline.

Pipeline step	**Corpus or procedure checkpoint**	**Use in the reference model**
2024 scoping review	PRISMA-guided review of five sources for 2020–2024 Neuro-Symbolic literature; 1,428 records extracted, 392 candidates screened in detail, 167 code-bearing papers identified, and 158 retained after full-paper screening [20].	Established the initial Neuro-Symbolic field map, search vocabulary, and thematic baseline: knowledge representation, learning and inference, explainability and trustworthiness, logic and reasoning, and meta-cognition.
2025 updated systematic review	Protocol-guided systematic review across nine sources through 23 May 2025; 5,497 records retrieved, 3,018 duplicates removed, 2,479 unique records screened, 1,365 on-topic records identified, and 1,304 eligible full-text Neuro-Symbolic records retained [19].	Defined the eligible Neuro-Symbolic literature corpus before applying the public-artifact and reproducibility gates.
Public-artifact gate	Repository verification removed 849 records with no verifiable public code artifact, leaving 455 code-bearing studies for attempted same-artifact rerun [19].	Separated claim-level Neuro-Symbolic papers from systems whose released materials could be inspected, rebuilt, and executed.
Same-artifact rerun gate	The attempted rerun pool produced 48 fully reproduced systems and 37 partially reproduced systems, yielding 85 verified functional systems from 455 attempted reruns [19].	Established the verified-system corpus used for reference-model derivation. O3, O4, and O5 outcomes were excluded from model construction.
Architectural extraction	Each verified system was decomposed into normalized fields: problem-set domain, operating environment, controller, neural component, symbolic store, reasoning process, system input/output, verification mechanism, architectural family, and primary Neuro-Symbolic level.	Converted and reproduced systems into comparable role-level records.
Cross-system synthesis	The normalized records were compared across the 85 verified systems for recurring roles, information flows, coupling patterns, verification structures, integration depth, and build dependencies.	Produced the Neuro-Symbolic reference-model roles, Level 0–7 ordinal scheme, six architectural families, and build-order argument.

reference model by applying a single repeated test to the normalized records to determine whether each retained system could be represented using the same role vocabulary without adding slots that existed only to accommodate one paper's terminology. When a system required an additional slot, the extraction vocabulary was revised and checked against the full verified corpus during reconciliation, so the final reference model reflects the minimum role set that covered the entire verified-system corpus.

3.7 Auditability and limits of inference

The reference model is descriptive of the Neuro-Symbolic systems identified from the included 85 studies of the reproducibility study, and the scope of the model's descriptive claims is bounded by the verified-system corpus from which the model was abstracted. The model describes recurring roles, coupling patterns, ordinal levels, and build-order dependencies observed across public-artifact Neuro-Symbolic systems that survived independent same-artifact rerun. The model does not claim coverage of proprietary systems whose code was never released, unreleased systems whose artifacts were withheld for legal or commercial reasons, or systems that may be reproducible in principle but could not be rerun under the same-artifact protocol within the constraints the audit imposed. A practitioner designing a system in a domain not represented among the 85 verified functional systems should treat the reference model as a grounded starting point. The extraction vocabulary was refined iteratively against the verified corpus, which means the final role set is fitted to the 85 systems from which it was derived and may require extension when applied to domains not represented in that corpus. The public-artifact bias the methodology introduces is intentional, but the bias does limit the scope of inference [19]. Grounding the reference model in systems whose operation could be inspected, rebuilt, and executed means the model rests on confirmed structural evidence, which strengthens construct validity and traceability at the cost of excluding systems that were never publicly executable. Failed reruns were not used to derive architectural roles or coupling patterns, but the 370 failed attempts across the 455 attempted reruns document the scale of the reproducibility deficit in the Neuro-Symbolic literature and justify why the functional-system gate was necessary. A model derived from all 1,304 eligible full-text records would mix verified structural evidence with unverified architectural claims in proportions that could not be separated after the fact. Examining what the 85 verified functional systems share in domain distribution, operating environment, verification regime, and ordinal level, therefore, reveals the current shape of functional Neuro-Symbolic AI as the field's own released and reproduced artifacts define it.

4 Corpus Overview of the 85 Verified Neuro-Symbolic Systems

4.1 Common Themes across the Corpus of Verified Neuro-Symbolic systems

Across the 85 independently verified systems, functional Neuro-Symbolic systems appear in many forms but still resolve to one shared architecture. The direct implementation of a Neuro-Symbolic system spans ordinal levels of *Neuro-Symbolicity* from Level 0 gated hybrid and Level 1 pipeline reasoner through Level 2 constraint-integrated, Level 3 iterative cooperative, and Level 4 co-optimized reasoner to the rarer Level 5 self-explaining and trust-aware systems and Level 6 meta-controlled systems. No primary Level 7, Self-Improving Neuro-Symbolic systems, were observed. The spread across the ordinal scale shows that *Neuro-Symbolicity* is a range of tighter or looser integrations built from the same roles grounded on the shared reference model.

4.2 High-level distribution of domains, environments, reasoning, and verification

Figure 2 summarizes the verified corpus by assigning each paper one dominant Neuro-Symbolic Problem-Set Domain, one dominant operating environment, one dominant verification regime, and one primary Neuro-Symbolic level. We define these dimensions to capture the main problem class that a Neuro-Symbolic system addresses, the setting in which it operates, the dominant method by which Neuro-Symbolic outputs are checked, and the position within the ordinal Neuro-Symbolic scheme. The distribution is broad in task coverage but concentrated in a few practical operating modes. Structured prediction and knowledge is the largest domain at 24 systems, benchmark and dataset-defined settings dominate the environment panel at 48, and empirical benchmark evaluation is the most common verification regime at 39. The level distribution is flatter but still bounded, with Levels 1 and 3 modal at 18 each, Levels 5 and 6 accounting for only three systems in total, and no primary Level 7 systems. These counts describe only the 85 systems that passed the functional gate, not the full literature, but they suggest that functional Neuro-Symbolic systems are presently easiest to realize where the environment is explicit and outputs can be measured, executed, or formally checked.

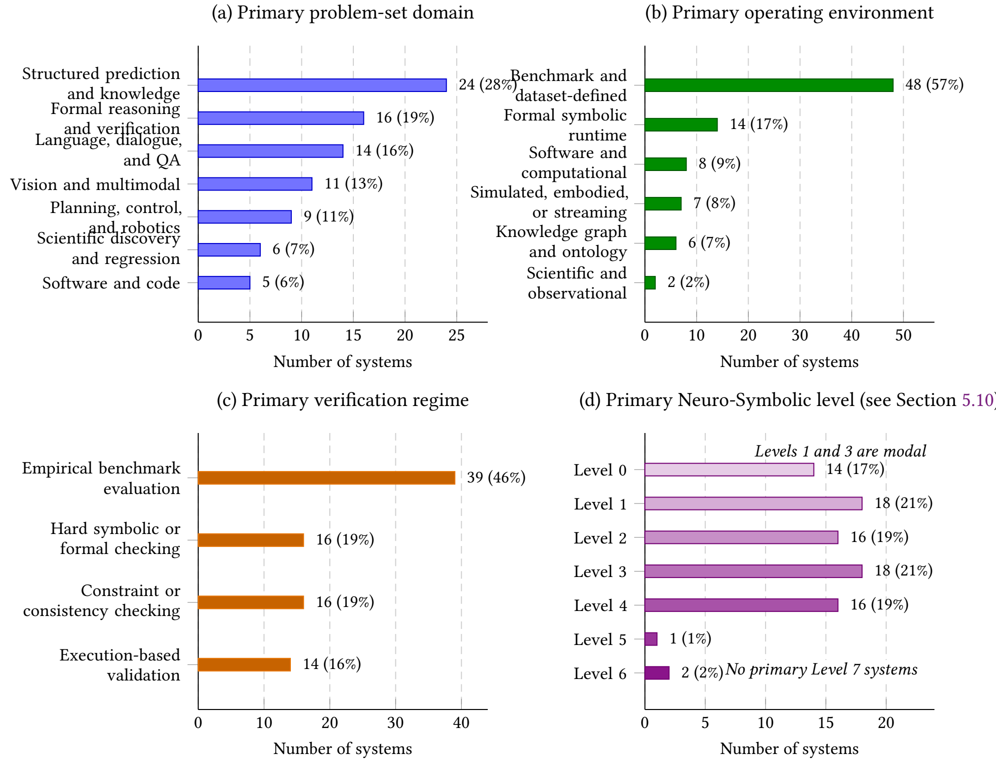

Fig. 2. **Corpus landscape of the 85 verified Neuro-Symbolic systems.** Each verified system was assigned one dominant problem-set domain, one dominant operating environment, and one dominant verification regime based on the normalized extraction fields used in the synthesis procedure. Panels (a)–(c) report those dominant per-paper assignments, while panel (d) reports the exact primary Neuro-Symbolic level distribution. Bar-end labels show counts and percentages with respect to $n = 85$.

5 Neuro-Symbolic Reference Model

5.1 Overview of Neuro-Symbolic Reference Model

Neuro-Symbolic systems, as described in Figure 3, work from the ground up, beginning with a Neuro-Symbolic Problem-Set Domain and a Neuro-Symbolic Environment before any internal Neuro-Symbolic loop can be meaningfully built. In Figure 3, the Problem-Set Domain defines the class of tasks, constraints, and knowledge structures that make the target problem meaningful, while the environment is the operative world in which the system receives inputs, maintains state, reasons, acts, and produces consequences. Within that setting, a Neuro-Symbolic Controller coordinates a Neural component that learns and proposes, a Symbolic Store that holds explicit knowledge and structured state, and a Reasoning Process that verifies, plans, proves, executes, or repairs across training and/or runtime. Figure 3 also distinguishes Neuro-Symbolic System Input and Neuro-Symbolic System Output from the broader flow of Input Data and Output Information, which keeps the system boundary explicit to limit collapsing the tasking, world state, and internal process into one pipeline. The ordinal levels of Neuro-Symbolic systems describe different degrees of *Neuro-Symbolicity* that appear in the literature while still deriving from the shared base model in Figure 3. Figure 4 describes the observed ordinal scale from Level 0 through Level 7 and shows the scale as

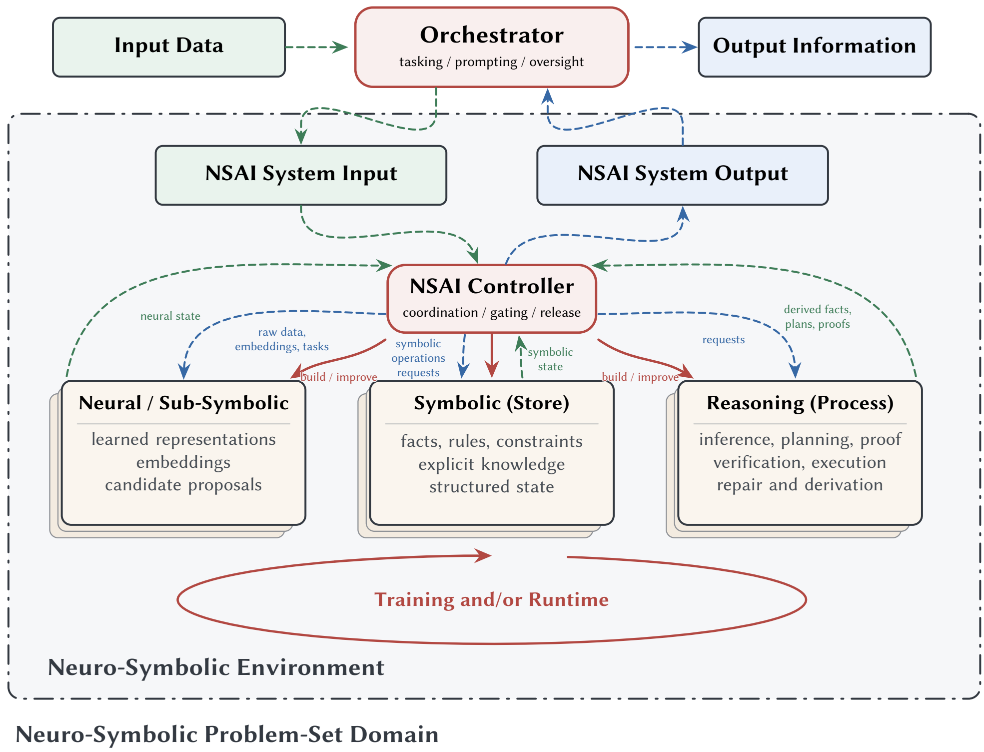

Fig. 3. **Neuro-Symbolic Reference Model.** The Neuro-Symbolic Reference Model defines a Neuro-Symbolic AI system as a situated system operating within a *Neuro-Symbolic Problem-Set Domain* and a *Neuro-Symbolic Environment*. The Problem-Set Domain specifies the class of tasks, constraints, and knowledge structures for which the system is designed, while the environment is the operative world in which the system functions, receives inputs, maintains state, reasons, acts, and produces consequences. Within this setting, a *Neuro-Symbolic Controller* coordinates the core Neuro-Symbolic process across training and/or runtime by orchestrating three principal components: a *Neural* component for learned representations, embeddings, perception, and candidate solution generation; a *Symbolic Store* for explicit knowledge, concepts, rules, constraints, and structured state; and a *Reasoning Process* for inference, proof, planning, verification, or symbolic execution. The model also distinguishes between *Neuro-Symbolic System Input* and *Neuro-Symbolic System Output* and the broader external flow of *Input Data* and *Output Information*. In some deployments, an external *Orchestrator* mediates this flow by providing tasking, prompting, workflow coordination, oversight, acceptance, or escalation and this role may be instantiated by a human, a software process, or another agent, and may be explicit or implicit depending on the system. The arrows indicate the flow of information and control through the architecture, while the build/improve links indicate that the neural and symbolic parts may both be refined over time.

an ordinal progression from the shared base in Figure 3. At the lower end, Level 0 gated hybrid and Level 1 pipeline reasoner keep the symbolic side relatively shallow or one-way. Middle levels, such as Level 2 constraint-integrated, Level 3 iterative cooperative, and Level 4 co-optimized reasoner, show stronger coupling and repeated Neuro-Symbolic interaction. Higher levels, such as Level 5 self-explaining and trust-aware, Level 6 meta-controlled, and the adopted Level 7 self-improving self-architecting extension place more verification, strategic control, and adaptation above the same core architecture. The verified corpus already spans primary Levels 0 through 6, with Level 7 being the next logical progression emerging from the ordinal scale that was not observed in the corpus

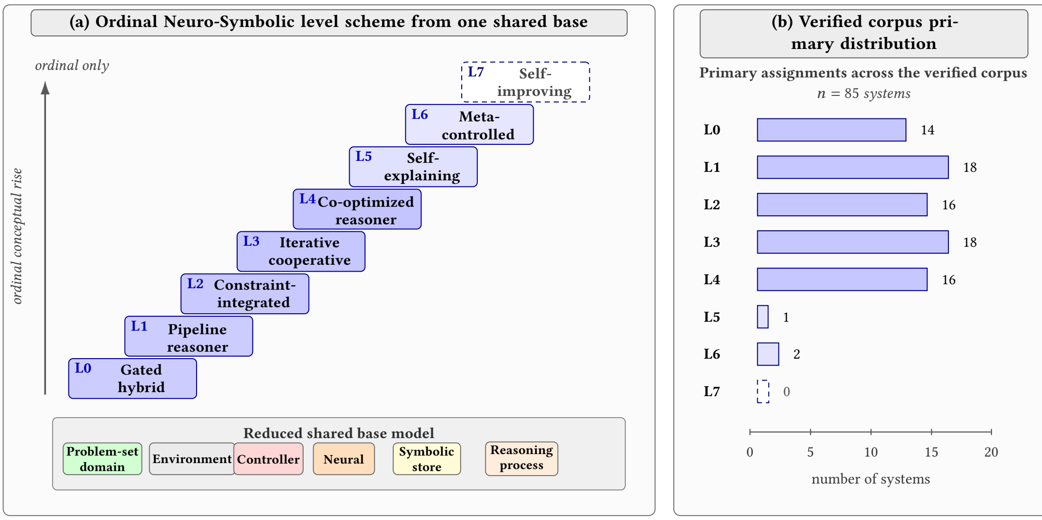

Fig. 4. **Neuro-Symbolic levels as an ordinal scheme plus empirical distribution.** Panel (a) presents the adopted Level 0–7 Neuro-Symbolic scheme as an *ordinal* progression (ordered conceptual level scheme) derived from one shared Neuro-Symbolic base. Panel (b) reports the exact primary-level counts in the verified corpus. Panel (a) illustrates the ordinal scale as progressive steps of Neuro-Symbolic integration and coupling, beginning at Level 0 gated hybrid, Level 1 pipeline reasoner, Level 2 constraint-integrated, Level 3 iterative cooperative through to Level 4 co-optimized reasoner, Level 5 self-explaining and trust-aware, Level 6 meta-controlled, and Level 7 self-improving self-architecting. Dashed marks at Level 7 indicate an adopted but unsubstantiated extension in the present verified set.

of verified Neuro-Symbolic literature. The required build order follows from the reference model because a functional Neuro-Symbolic system cannot be assembled by starting with neural and symbolic modules alone. As described by Figure 6, the build order requirement of functional Neuro-Symbolic systems begins with an effective Neuro-Symbolic Problem-Set Domain as the ground, a Neuro-Symbolic Environment as the foundation, and then the three sub-components, including the Neural component, Symbolic Store, and Reasoning Process, with the Neuro-Symbolic Controller as the structure built across them. The build order for a Neuro-Symbolic system is critical because the wrong Problem-Set Domain removes symbolic correctness, and a weak environment leaves no operative world in which inputs, state, actions, feedback, and consequences can be grounded. Only after those outer layers are sound can the inner pillars become functional, stable, and useful as a system. We therefore begin the construction of a Neuro-Symbolic system with exploration and discovery of an appropriate Neuro-Symbolic Problem-Set Domain that can act as a solid bedrock from which a strong Neuro-Symbolic Environmental foundation can be developed.

5.2 Data Flow and Data Interfaces

As shown in Figure 3, the reference model can be read as a set of interfacing data-contracts. Each arrow specifies what is admitted, transformed, checked, or released across the Neuro-Symbolic loop. At a minimum, each payload should preserve task identity, domain binding, environment binding, provenance, and status.

5.2.1 Input Data → Orchestrator. This arrow carries raw external artifacts and source metadata into workflow control. Its output is an intake record, not yet a valid internal NSAI task.

5.2.2 Orchestrator → NSAI System Input. This arrow converts intake material into a typed task package containing the goal, domain binding, environment binding, constraints, success criteria, and resource limits.

5.2.3 NSAI System Input → NSAI Controller. This arrow admits the task into the internal loop. Its immediate output is a controller decision to accept, reject, decompose, or request clarification.

5.2.4 NSAI Controller → Neural / Sub-Symbolic. This arrow sends task context, observations, and target representation constraints to the Neural component. The purpose is to request a candidate proposal in a form that can later be grounded, stored, and checked.

5.2.5 Neural / Sub-Symbolic → NSAI Controller. This arrow returns candidate answers, actions, programs, proofs, or representations together with confidence, uncertainty, and model provenance. These outputs remain provisional until they are symbolically grounded and checked.

5.2.6 NSAI Controller → Neural / Sub-Symbolic (build/improve). This arrow carries supervision, counterexamples, rewards, losses, or refinement signals to improve proposal quality. Its output is an updated model state or deployable model version.

*5.2.7 NSAI Controller → Symbolic (Store) (*symbolic operations requests*).* This arrow issues read, write, query, type-check, or constraint-application requests over explicit symbolic memory. This is the main interface for manipulating stored facts, rules, constraints, and structured state.

5.2.8 Symbolic (Store) → NSAI Controller. This arrow returns an authoritative symbolic state, including facts, constraints, typed objects, state deltas, or conflict reports. The Controller should treat this as the explicit state basis for later reasoning.

5.2.9 NSAI Controller → Symbolic (Store) (build/improve). This arrow updates the symbolic substrate itself, for example, by revising ontologies, schemas, rules, or indexes. Its output should be a versioned symbolic update with consistency status.

*5.2.10 NSAI Controller → Reasoning (Process) (*requests*).* This arrow invokes proof, planning, checking, execution, search, or repair over explicitly identified symbolic objects. The request should specify both the reasoning objective and the symbolic state to be used.

5.2.11 Reasoning (Process) → NSAI Controller. This arrow returns derived facts, plans, proofs, execution traces, counterexamples, or repair diagnoses. These outputs should carry explicit status such as verified, failed, inconsistent, or needs repair.

5.2.12 NSAI Controller → Reasoning (Process) (build/improve). This arrow refines the reasoning stack itself, for example, through tactic updates, heuristic changes, validator changes, or solver reconfiguration. Its output is an updated reasoning capability.

5.2.13 NSAI Controller → NSAI System Output. This arrow packages only validated results for release. The output should include the result, its validation basis, provenance, and any residual-risk or review-required flags.

5.2.14 NSAI System Output → Orchestrator. This arrow returns the checked internal result to external mediation. It should carry both the result and the evidence needed for delivery, hold, retry, or escalation.

5.2.15 Orchestrator → Output Information. This arrow converts the checked result into externally consumable output. Presentation may change here, but validation status and provenance should be preserved.

5.2.16 Training and/or Runtime loop. The training/runtime band in Figure 3 denotes that the same interfaces remain valid across both phases. Training adds corrective and update signals, while runtime adds release, latency, rollback, and escalation conditions.

5.3 Outer layer I: Neuro-Symbolic Problem-Set Domain

The Neuro-Symbolic Problem-Set Domain is the load-bearing ground of a Neuro-Symbolic system because it defines the class of problems, the symbolic objects, and the success conditions that give meaning to both the external data flow and the internal loop. In Figure 3, raw Input Data and Output Information sit in the broader external flow and may come from outside the Neuro-Symbolic Environment before an Orchestrator or interface turns them into Neuro-Symbolic System Input and later maps Neuro-Symbolic System Output back into usable output information. The domain, therefore, sits at the same outer level of description as that external flow. The Neuro-Symbolic Problem-Set Domain fixes which incoming data are relevant to the task, what symbolic form candidate solutions must take, and what must be true for an output to count as correct. Figure 2 provides a descriptive distribution of dominant Problem-Set Domains, and examples from the verified corpus include AlphaGeometry, which is an informal geometry problem that only becomes meaningful because the domain is Euclidean theorem proving, so the system must produce constructions and proofs that a deductive engine can check [79]. In NSGRAPH, a graph image and a natural-language question are admitted into a visual graph question answering domain, where recovered graph facts must support an ASP-derived answer [29]. A-NeSI and Neural Symbolic Model for Space Physics show the same dependence in more formal and scientific settings, where raw inputs only become usable once the domain fixes worlds, beliefs, outputs, formulas, units, and acceptable verification traces [82, 89]. The design consequence is that a Neuro-Symbolic system cannot be built soundly by starting with modules before the ground has been set. Figure 6 makes the construction metaphor explicit by placing the Neuro-Symbolic Problem-Set Domain below the Neuro-Symbolic Environment, then the three pillars, then the Neuro-Symbolic Controller. If that ground is weak, external data cannot be converted into a well-posed system input, output information cannot be judged against a stable target, and the Symbolic Store and Reasoning Process loses the predicate that tells them what to verify, reject, or repair. A well-specified domain, therefore, does for Neuro-Symbolic systems what firm ground does for a physical structure.

5.4 Outer layer II: Neuro-Symbolic environment

Figure 3 presents the Neuro-Symbolic Environment as the world that encloses the Neuro-Symbolic system. The Neuro-Symbolic Environment is the layer wherein input data may enter the Neuro-Symbolic system to be transformed into a Neuro-Symbolic system input, and where the current world state is maintained. Figure 6 presents the same layer as the foundation poured into the Neuro-Symbolic Problem-Set Domain bedrock, beneath the three pillars and the controller roof. Figures 3 and 6 present the environment as both the world in which the system operates and the foundation that carries everything built above it. A-NeSI provides an explicit example of a Neuro-Symbolic Environment through a world, belief, and output setting governed by symbolic function c, while AlphaTrans treats repositories, ASTs, builds, and translated tests as the operative software world in which candidate translations succeed or fail [40, 82]. Further evidence from the verified corpus shows that functional Neuro-Symbolic systems are concentrated in environments that are sufficiently explicit to yield grounded checks, even though those environments vary widely in form. Figure 2 shows benchmark- and dataset-defined settings as the dominant operating environment, with 48 systems, followed by formal symbolic runtimes with 14 systems, with the remaining systems spread across software and computational, simulated, embodied, or streaming, knowledge graph and ontology, and scientific and observational settings. Explicit examples from the literature include NSGRAPH which operates over CLEGR-derived graph images where the recovered graph becomes the world that ASP reasons over, A-NeSI which uses a formal world-belief-output setting with symbolic pruning, and PhyE2E which works over observational scientific data and symbolic regression benchmarks where physical priors and unit systems are part of the environment itself [29, 82, 89]. The roles of the neural, symbolic, and reasoning components persist across these Neuro-Symbolic systems, but the Neuro-Symbolic sub-components only become functional when the world around them is stable enough to be observed, updated, executed, or checked. Treating the environment as a true foundation has a direct design consequence, because sub-component design/selection, controller design, and system evaluation can only be specified after the world of operation has been well defined. When the Neuro-Symbolic Environment foundation is weak or not well defined, the overall effectiveness of the Neuro-Symbolic system design is degraded because the environment determines which sub-components are needed and how they can be integrated (the three sub-component pillars require a strong foundation to rest upon), which then allows the Neuro-Symbolic Controller to be effectively built.

5.5 Core proposal layer: Neural component

The Neural component is the proposal layer of a functional Neuro-Symbolic system because the neural component turns raw or high-dimensional Neuro-Symbolic System input into learned representations and candidate symbolic objects that the rest of the architecture can store, inspect, and reason over. In Figure 3, this layer produces candidate answers, steps, programs, proofs, actions, or constructions while the Symbolic Store preserves explicit state and the Reasoning Process verifies, derives, repairs, or executes against that state. AlphaGeometry is one of many examples that describe an effective neural component wherein a transformer language model trained on synthetic theorem-proof data proposes auxiliary constructions and the formal geometry engine proves or rejects them [79]. The Neural handoff is the interface constraint that makes the neural side usable within a Neuro-Symbolic system, because a proposal only becomes functional when it crosses into a form that can be checked by another component of the Neuro-Symbolic system.

5.6 Core explicit knowledge layer: Symbolic store

The Symbolic Store gives a functional Neuro-Symbolic system persistent explicit knowledge and symbolic state, so the system can carry proposals, constraints, and intermediate results forward into checking and repair. Figure 3 places this store beside the Neural component and the Reasoning Process because the system must preserve rules, facts, constraints, ontologies, programs, proofs, graphs, and typed state in a form that later operations can read, update, and reuse across training and/or runtime. NSGRAPH shows this role directly, because it stores recovered graph facts, parsed question programs, and an ASP theory before ASP graph reasoning computes the answer [29]. The store gives the system a stable object of verification and a memory that survives more than one step. When designers leave the store implicit, facts and constraints stay buried in transient activations or one-shot strings, the controller loses a stable symbolic state, and verification degrades into shallow checking.

5.7 Core inference layer: Reasoning process

The Reasoning Process makes a functional Neuro-Symbolic system more than a generator because it performs explicit inference and checking over symbolic representations and decides whether a candidate is valid, derivable, executable, or in need of repair. Figure 3 places the Reasoning Process beside the Neural and Symbolic Store components because the reasoner acts on stored rules, facts, constraints, programs, proofs, and symbolic state instead of producing answers from raw input alone. The reasoning component carries out symbolic actions such as proof search, planning, constraint solving, symbolic execution, or repair, and these operations create the verification step that tells the system whether a proposal survives. CodePlan demonstrates an effective reasoner component wherein the system builds a repository-wide dependency graph and a plan graph, performs change-impact analysis to infer new edit obligations, and uses oracle checks on repository state to decide whether another planning round is required [11]. Explicit reasoning improves reliability, but it also tightens the interface among proposals, stored structure, and the operative world, because planning, search, or symbolic execution only work when the Neural component emits typed symbolic objects and the Symbolic Store preserves the facts and constraints those operations require. When designers pass free-form text, poorly grounded symbols, or incomplete state into the Reasoning Process, variable binding breaks, search drifts, and checking collapses into heuristic scoring. Proposals, symbolic state, checks, repairs, and outputs thus are more effective when they are allowed to move through training and runtime as one connected operational loop.

5.8 Core orchestration: Neuro-Symbolic controller

The Neuro-Symbolic Controller is the functional spine of a Neuro-Symbolic system because it sequences and gates neural proposals, symbolic state updates, reasoning checks, and environment interaction into one operating loop. Figure 3 places the controller inside the Neuro-Symbolic Environment above the Neural component, the Symbolic Store, and the Reasoning Process because the controller decides when the system proposes, when it commits symbolic state, when it invokes symbolic operations, and when a checked result can leave the loop as Neuro-Symbolic System Output. The external Orchestrator serves a different role because it handles tasking, prompting, workflow coordination, oversight, acceptance, or escalation outside the system boundary. AdaLoGN demonstrates an effective controller by building a raw Text Logic Graph, proposing symbolic extensions, admitting only relevant extensions through a learned scorer, rerunning graph message passing, and rescoring answer options under an explicit iterative loop [58]. The controller, therefore, becomes a functional bottleneck because stable stopping criteria, safe state management, and reliable release conditions all pass through it. When control logic is weak, the system can admit irrelevant proposals, overwrite useful symbolic states, invoke reasoning at the wrong time, or emit output before verification has converged. Input admission, output release, verification feedback, and adaptation signals must therefore pass through the controller in a well-defined operational flow.

5.9 Operational flows: inputs, outputs, training/runtime, verification, and adaptation

Operational flows make the reference model functional by specifying how the controller admits input, commits output, separates training from runtime, and uses verification feedback to improve later behavior. In Figure 3, Input Data stays outside the system until an orchestrator or interface grounds it as Neuro-Symbolic System Input inside the Neuro-Symbolic Environment, and Neuro-Symbolic System Output stays inside the system until the controller releases a checked result that can become external Output Information. Figure 3 also marks training and/or runtime as one shared operational band, which means the architecture does not assume a single learning schedule, only that proposals, symbolic state, reasoning, and control preserve the same interfaces whether the system is learning, executing, or doing both. Examples of effective feedback paths include Closed Loop Neuro-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning because the system parses a neural proposal into a grammar-constrained symbolic form, executes it symbolically, diagnoses failure, and feeds the correction back into training as a pseudo-label [57]. Verification and adaptation, therefore, do not sit outside the system as after-the-fact evaluation, but instead constrain what the controller allows to survive and determine whether the next pass should accept, repair, retry, or improve the inner components, which is the same operational pressure that later separates weaker hybrids from more tightly integrated and therefore higher levels of Neuro-Symbolic systems.

5.10 Levels of Neuro-Symbolic systems and how they derive from the same base model

As described in Figure 4, all functional Neuro-Symbolic systems in the verified corpus follow the common structure of the Neuro-Symbolic Reference Model in Figure 3, but differences in integration strength, reasoning depth, feedback structure, controller involvement, and meta-control place them at different points on an ordinal scale of Neuro-Symbolicity. At Level 0, the gated hybrid level, the symbolic structure mainly filters, constrains, or repairs a largely neural flow. CORRPUS is a Level 0 example [28], where code-structured state supports story tracking but prompt-mediated generation still does most of the work, and Error Detection and Constraint Recovery in HMC [51], where hierarchy constraints detect and repair violations after prediction. Level 1 pipeline reasoner adds a clearer one-way handoff from neural proposal to symbolic reasoning. An example of a level 1 Neuro-Symbolic system is LINC [70], where an LLM parses natural language into first-order logic for Prover9, and Embed2Sym [9], where learned embeddings are lifted into ASP facts and optimized symbolically. Level 2 constraint-integrated systems move symbolic structure into learning or scoring itself. An example of a level 2 Neuro-Symbolic system is CL-STE [87], where clause satisfaction enters the loss through the straight-through estimator, and Logic Tensor Networks [10], where differentiable logical constraints shape learned predicate values directly. The middle of the ladder is where verified systems begin to run repeated Neuro-Symbolic loops. Level 3 iterative cooperative systems alternate proposal, checking, and revision during inference. An example of a level 3 Neuro-Symbolic system is Dynamic Planning with an LLM [21], where neural goal and belief completion repeatedly interact with symbolic PDDL planning, and Xander [73], where best-first search, symbolic partial-query checking, execution, and repair sit inside the text-to-SQL loop. Level 4 co-optimized reasoner systems tie the neural and symbolic sides together around reasoning success itself. An example of a level 4 Neuro-Symbolic system is NeuPSL [74], where neural predicates and weighted logical clauses meet in joint convex inference, and DeepStochLog [83], where neural probabilities live inside a stochastic logic program whose logic-driven inference defines the computation. Level 5 self-explaining and trust-aware systems treat explanations and critique traces as part of the reasoning object. Examples of a level 5 Neuro-Symbolic system are NS-NLI [34], where introspective natural-logic revision operates over explicit reasoning paths, and PEIRCE [75], where critique and explanation artifacts become targets of symbolic revision. The top of the ladder depends on explicit strategic control over the Neuro-Symbolic loop itself, and Figure 4 shows how rare that remains in the verified corpus. Level 6 meta-controlled systems place a higher-order control policy above proposal and checking. An example of a level 6 Neuro-Symbolic system is Think before You Simulate [42], where a causal module decides when neural simulation is needed and where it should enter the loop, and Neuro-Symbolic Repair of Test Flakiness [17], where an inspector-generator-validator cycle selects repair actions under explicit feedback. Level 7 self-improving self-architecting has no verified examples in the present corpus, because no reproduced system yet diagnoses its own architectural gaps, builds or revises its own neural, symbolic, or reasoning modules, and then verifies those changes inside the same functional boundary. Figure 4 shows that Levels 1 and 3 are the modal assignments in the verified set, and also shows that Level 7 remains a hypothesis about the next step, one that would require stronger environmental support, tighter verification hooks, and stable adaptation loops that current systems do not yet sustain. No-op roles constrain level assignment. A no-op controller (e.g., a fixed pass-through pipeline), a no-op reasoning process (e.g., a solver that is invoked but returns a fixed result), or a no-op symbolic store (e.g., a static lookup table that is never updated) may satisfy structural presence but does not count as evidence for a higher Neuro-Symbolic level. Higher levels require active feedback, verification, repair, co-optimization, or meta-control.

5.11 Minimum conditions for a functional Neuro-Symbolic system

A system may be observed as functionally Neuro-Symbolic when the system contains (1) a Problem-Set Domain that fixes the task and what counts as a valid solution, (2) an operating environment that provides inputs, state, and consequences, (3) a controller that coordinates the overall process, (4) a neural component that learns or proposes candidate solutions, (5) an explicit symbolic store that holds structured knowledge or symbolic state, and (6) a reasoning process that verifies, derives, or repairs candidate results. A required Neuro-Symbolic role may be instantiated as a no-op (no-operation) when the role is present in the released workflow, has a defined input-output interface, is reachable during the reproduced run, and returns an identity, pass-through, empty, or fixed result without materially transforming system state in the evaluated configuration. A no-op role is distinct from an absent role, as an absent role has no observable implementation or interface, whereas a no-op role preserves the architectural contract while contributing no substantive operation in that run. A no-op role may satisfy the structural presence requirement only when the overall system still contains an active Neuro-Symbolic coupling. A system in which the neural component, symbolic store, and reasoning process are all no-op is not a functional Neuro-Symbolic system under definition 1.1. A required role may be active or no-op, but it may not be absent. Active roles materially transform, derive, verify, repair, or coordinate system state, while no-op roles preserve a required interface while passing through or returning a fixed result in the evaluated configuration.

6 Architectural Families of Functional Neuro-Symbolic Systems

6.1 Neural-to-symbolic lifting pipelines

Neural-to-symbolic lifting pipelines work by enforcing a clean one-way interface in which a neural module converts raw data into explicit symbols, and a symbolic engine then carries the main burden of reasoning or execution. As Figure 5 shows, the neural side lifts images, text, or learned embeddings into a typed artifact such as facts, formulas, or programs, commits that artifact to the Symbolic Store, and then hands control to the Reasoning Process, which operates on the symbolic object. NSGRAPH exemplifies an effective Neural-to-symbolic lifting pipeline by lifting a graph image and a question into graph facts and an ASP question program, after which ASP computes the answer over the recovered structure [29]. The same clean boundary makes interface quality the dominant failure point, because the solver can inspect and execute only the lifted artifact and cannot repair missing entities, incorrectly transcribed labels, or mis-formed relations once they enter symbolic state as ground truth. Pressure to reduce that brittle handoff is what motivates designs in which symbolic structure shapes learning before the final lift. To illustrate this family beyond the verified corpus, consider diagrammatic reasoning as a canonical toy problem that this family handles effectively, in which the Problem-Set Domain is fixed as the class of structural queries over diagram content, the Environment is defined by the space of renderable diagram instances, the formal query language that specifies what structural properties must hold, and the execution context that returns grounded verification outcomes against which every candidate symbolic interpretation can be accepted or rejected, and the three core pillars are instantiated by a neural perception module that interprets the raw image and lifts its visual content into an explicit symbolic representation such as typed nodes, labelled edges, and spatial relations committed to the Symbolic Store, and a Reasoning Process that executes queries, checks constraints, or derives conclusions over that recovered structure without any further access to the original image.

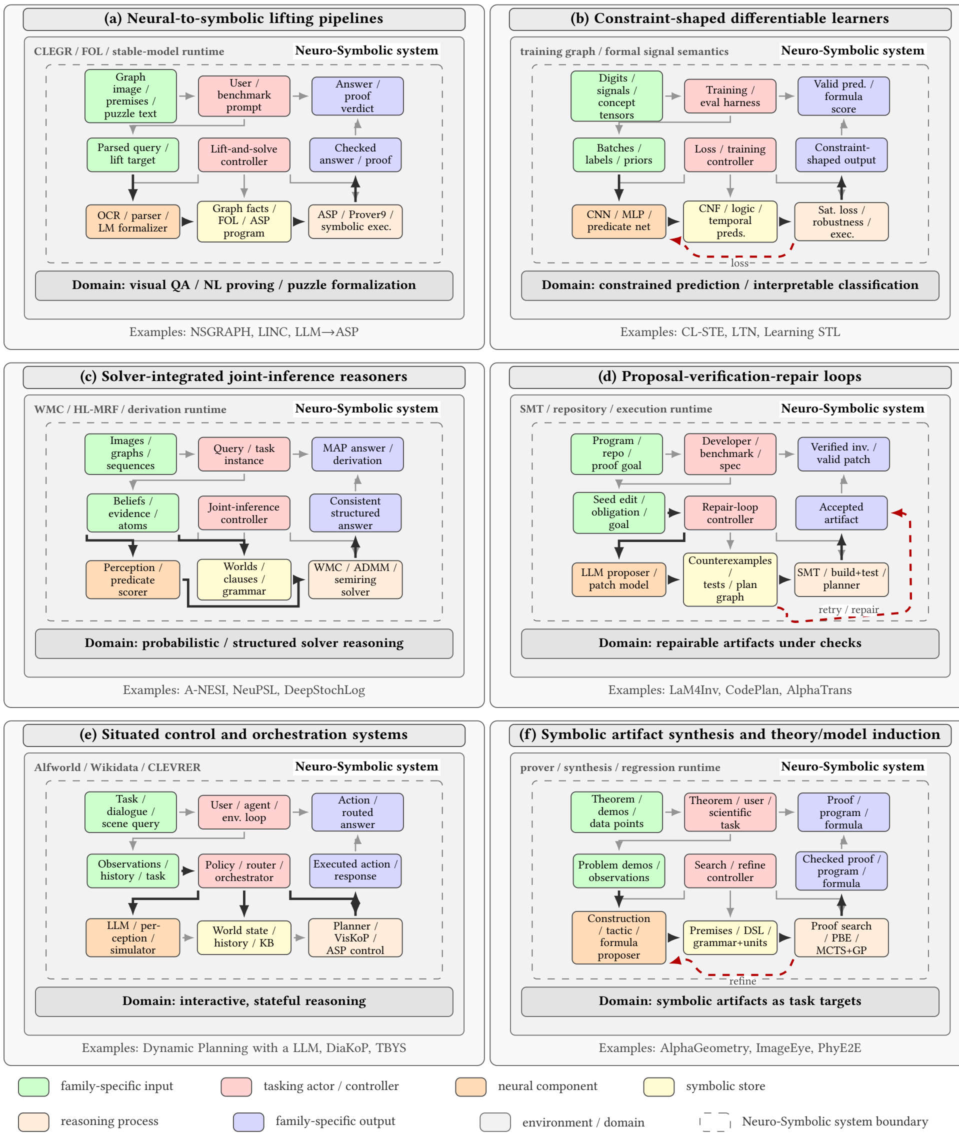

Fig. 5. **Six architectural families of functional Neuro-Symbolic systems.** Each panel is a role-preserving projection of the Neuro-Symbolic reference model in Fig. 3. All six panels retain the same standardized scaffold and the same component layout, including a family-specific external input, external tasking actor, and external output, a shared environment and problem-set domain, a Neuro-Symbolic system boundary, and, inside that boundary, family-specific system input, controller, system output, neural component, symbolic store, and reasoning process. The darker arrows mark the dominant coupling pattern for each family, while the lighter arrows preserve the common reference-model structure. Representative systems listed at the bottom of each panel demonstrate how the abstracted families correspond to real Neuro-Symbolic systems found in the verified corpus.

6.2 Constraint-shaped differentiable learners

Constraint-shaped differentiable learners work by putting symbolic structure into the learning signal so it shapes neural optimization. Figure 5 demonstrates an effective Constraint-shaped differentiable learner in which the Neural component produces candidate values, the Symbolic Store supplies clauses, formulas, or predicates, and the Reasoning Process turns that stored structure into satisfiability, robustness, or consistency objectives that the Controller feeds back during training or scoring. CL-STE instantiates the family by converting propositional CNF constraints and known facts into clause-wise deduction and satisfaction losses passed through a straight-through estimator, so the predictor is updated toward constraint-compatible assignments during optimization [87]. Compared with a pure lifting pipeline, this design reduces dependence on a single post hoc interface contract, but it introduces a surrogate mismatch because improving a soft logical objective does not guarantee the globally consistent discrete solution that a downstream task may require. To illustrate this family beyond the verified corpus, consider computational chemistry and synthesis as a canonical toy problem that this family handles effectively, in which the Problem-Set Domain is fixed as the class of chemically valid molecular structures and reaction pathways whose correctness is expressible in terms of symbolic chemical laws, the Environment is defined by the space of possible molecular configurations, the thermodynamic and kinetic conditions under which reactions occur, and the experimental or simulation outcomes that determine whether a predicted structure or reaction is physically realizable, and the three core pillars are instantiated by a neural model that predicts molecular properties or reaction outcomes, a Symbolic Store that encodes chemical valence rules, bond-type restrictions, and reaction feasibility conditions, and a Reasoning Process that converts those constraints into satisfiability objectives injected into the learning signal so that the optimizer is steered toward chemically valid predictions during training rather than relying on a post hoc symbolic filter to reject invalid structures after the fact.

6.3 Solver-integrated joint-inference reasoners

Solver-integrated joint-inference reasoners solve a coupled problem by making a symbolic solver compute the final output from neural evidence and explicit symbolic structure. Figure 5 depicts the family as a joint solve in which the Neural component supplies beliefs, scores, or grounded atoms, the Symbolic Store holds worlds, clauses, or rules, and the Reasoning Process performs the optimization, derivation, or aggregation that returns one consistent structured answer under controller coordination. SLASH exemplifies an effective Solver-integrated joint-inference reasoner by turning neural probabilistic predicates into symbolic choices inside an ASP program and then using stable-model reasoning with probabilistic query aggregation to determine the prediction itself [77]. Because the solver now defines the prediction, the architecture gains global consistency but becomes sensitive to solver cost and abstraction mismatch, since incomplete rules or miscalibrated neural scores can yield an internally coherent answer that still misses the task state. When one coupled solve cannot recover from those errors, systems move toward explicit proposal, verification, and repair loops. To illustrate this family beyond the verified corpus, consider entity resolution for open-source intelligence as a canonical toy problem that this family handles effectively, in which the Problem-Set Domain is fixed as the class of globally consistent entity assignments over heterogeneous source mentions whose correctness is expressible in terms of co-reference, uniqueness, and transitivity constraints, the Environment is defined by the live corpus of heterogeneous intelligence sources, the identity graph over candidate entities, and the verification context that determines whether a proposed assignment violates any known constraint or conflicts with established ground truth, and the three core pillars are instantiated by a neural component that scores the likelihood that mentions refer to the same real-world entity, a Symbolic Store that holds co-reference rules, uniqueness constraints, and transitivity conditions over the candidate entity graph, and a joint solver that aggregates those neural scores under the symbolic constraints to produce a globally consistent entity assignment that no purely neural model could guarantee.

6.4 Proposal-verification-repair loops

Proposal-verification-repair loops are defined by explicit cycles that turn neural candidates into accepted artifacts only after symbolic checks reject, diagnose, and redirect failing proposals. Figure 5 depicts the family as a Neuro-Symbolic Controller that asks the Neural component for a patch, tactic, or proof step, sends the candidate together with symbolic state from the Symbolic Store to the Reasoning Process, and uses the returned test result, counterexample, or failed obligation to trigger another proposal. The Neuro-Symbolic system developed for Proving Olympiad Inequalities demonstrates an effective Proposal-verification-repair loop because an LLM proposes rewriting tactics and goal rankings, symbolic filters and counterexample checks prune bad branches, and final Lean verification accepts only proof sequences that satisfy the formal goal [61]. The tradeoff is that stronger verification improves trust only when validators expose actionable failure signals and the controller enforces stopping rules, since weak diagnostics can keep the loop generating new candidates without reducing the remaining error. When the same proposal-check-repair pattern must also manage observations, state updates, and actions in an external world, the architecture grows into situated control and orchestration. To illustrate this family beyond the verified corpus, consider cyber offense, cyber defense, and software development as canonical toy problems that this family handles effectively, in which the Problem-Set Domain is fixed as the class of symbolically verifiable artifacts whose correctness is expressible in terms of security policies, type contracts, formal specifications, or executable test conditions, the Environment is defined by the target system or codebase, the toolchain of sandboxes, static analyzers, formal checkers, and test suites that return grounded verification outcomes, and the consequences that follow from deploying an artifact that violates any checked condition, and the three core pillars are instantiated by a neural component that proposes candidate exploits, patches, or code edits, a Symbolic Store that holds vulnerability signatures, security policies, type contracts, and static analysis rules, and a Reasoning Process that executes each candidate against the verification toolchain to return a counterexample, a failed assertion, or a policy violation that the controller uses to reject the proposal and prompt a revised attempt, with the loop continuing until a candidate survives all symbolic checks or a stopping condition is reached.

6.5 Situated control and orchestration systems

Situated control and orchestration systems shift the core difficulty from producing a single correct artifact to managing action, state, and feedback inside an environment through an explicit controller. Figure 5 depicts the family as a Neuro-Symbolic Controller that receives observations and history from the environment, decides when the Neural component should interpret new inputs or propose actions, updates the Symbolic Store with world state and task context, and invokes the Reasoning Process for planning, routing, or execution before the system commits a response or action. ANSR-DT demonstrates an effective situated control and orchestration system because sensor streams feed a layered loop in which CNN-LSTM perception generates symbolic facts, rule-based reasoning updates decision state, and PPO adaptation revises policy and rule use, with sensor feedback and rule confidence thresholds closing the loop over detections, explanations, and actions [37]. The gain is a tighter grounding in consequences, but the cost is a stronger dependence on state abstraction and feedback quality, because partial or stale signals can leave the controller coordinating correctly over the wrong symbolic state. When the goal shifts from managing interaction to emitting a durable proof, program, or formula, the architecture moves toward symbolic artifact synthesis.

To illustrate this family beyond the verified corpus, consider embodied, situational, and robotic reasoning as canonical toy problems that this family handles effectively, in which the Problem-Set Domain is fixed as the class of goal-directed action sequences whose correctness is expressible in terms of task completion conditions, safety constraints, and physical feasibility requirements, the Environment is defined by the physical or simulated world that the agent inhabits, the sensor modalities through which it observes state, the action space through which it effects change, and the consequence structure that determines whether a committed action moves the agent toward or away from its goal, and the three core pillars are instantiated by a neural component that interprets raw sensor streams such as camera feeds, lidar returns, and proprioceptive signals into perceptual representations, a Symbolic Store that maintains a world model of object locations, agent goals, task constraints, and action history, and a Reasoning Process that executes a planner or policy over that symbolic state to select the next action, with the controller closing the loop by updating the world model from environmental feedback after each action is committed.

6.6 Symbolic artifact synthesis and theory/model induction systems

Symbolic artifact synthesis and theory or model induction systems are defined by producing an explicit symbolic artifact that can be executed, checked, and reused as the main system output. Figure 5 places the family where the Neural component proposes constructions or fragments, the Symbolic Store supplies premises, templates, grammars, or constraints, and the Reasoning Process searches or refines until the Controller can release a checked proof, program, formula, model, or theory that the environment can run again independently. The Neuro-Symbolic system described in *Making sense of raw input* demonstrates an effective symbolic artifact synthesis and theory/model induction system by combining a recognizer with joint SAT and ASP search to induce a Datalog-like theory from raw temporal observations, with synthesis guided by explanatory fit, low description cost, and unity conditions, and with candidate theories accepted only when the joint solve satisfies the observation and constraint templates [31]. The main constraint is search burden under a chosen symbolic abstraction, because weak templates or incomplete constraints can leave the verifier unable to distinguish plausible from reusable artifacts. The family, therefore, shows how the same controller, neural component, symbolic store, and reasoning process realize different Neuro-Symbolic levels by changing what must be synthesized and what the environment can check. To illustrate this family beyond the verified corpus, consider the discovery of physical laws from observational data by an agentic scientist as a canonical toy problem that this family handles effectively, in which a neural component proposes candidate functional forms over raw measurements, the Symbolic Store holds physical units, dimensional constraints, and conservation laws that confine the search space to physically meaningful candidates, and the Reasoning Process validates each candidate against held-out observations, feeding failed checks back to the controller until a symbolic law survives every constraint and is released as an independently verifiable artifact that any simulator or experiment can reuse. Kepler's derivation of elliptical orbits from Brahe's Mars observations [32, 47] is precisely this process, with the Problem-Set Domain fixed as physically verifiable orbital laws, the Environment defined by Brahe's measurement corpus and empirical checks, and the three pillars instantiated by Kepler himself, proposing candidate forms, rejecting each against observational residuals, and converging only when every constraint was satisfied, with the difference that a Neuro-Symbolic system replaces decades of manual search with a neural proposer, tacit geometric intuition with a formally complete Symbolic Store, and individual scientific judgment with a Reasoning Process that validates at a scale and speed no single scientist could sustain.

6.7 The Six Architectural Families in a Nutshell

The six Neuro-Symbolic Architectural families differ by their dominant coupling pattern, that is, by how the neural, symbolic, and reasoning components connect and what that connection is responsible for producing. In a *neural-to-symbolic lifting pipeline*, neural perception creates symbols and symbolic reasoning does the main work, making the quality of that handoff the single point of failure on which the entire system depends. In a *constraint-shaped differentiable learner*, symbolic knowledge shapes the neural model's training behavior directly, steering learning toward valid outputs before any inference takes place. In a *solver-integrated joint-inference reasoner*, neural scores and symbolic rules are fused inside one inference process, so neither side produces the answer alone, and both must be well-calibrated for the result to be correct. In a *proposal-verification-repair loop*, neural candidates are repeatedly verified and corrected by symbolic checks, cycling until a candidate passes or a stopping condition is reached, which means the usefulness of the loop depends entirely on how informative the symbolic feedback is. In a *situated control and orchestration system*, the system manages state, actions, feedback, and consequences over time inside a live environment. In a *symbolic artifact synthesis and theory induction system*, the goal is to produce a durable symbolic output such as a proof, program, formula, or theory that can be verified and reused independently of the system that produced it. The same underlying components appear across all six families and the coupling pattern is what determines which of the six Architectural Families a Neuro-Symbolic system most aligns with.

7 Discussion

7.1 Why Neuro-Symbolic Is Hard in Practice

The central practical difficulty in the development of a Neuro-Symbolic System is the identification of a Neuro-Symbolic Problem-Set Domain that is genuinely suitable for symbolic verification (and many domains are not practically solvable with Neuro-Symbolic systems), before one can construct the Neuro-Symbolic Environment in which such a Neuro-Symbolic System can operate. A viable domain must define a set of solvable symbolic problems, explicit success conditions, and symbolic objects that enable the Reasoning Process to distinguish a valid result. The environment must then expose the observations, state, actions, constraints, feedback, and consequences needed to test those conditions in an operative world. Figure 6 makes this build order explicit by placing the domain and environment beneath the Neural component, Symbolic Store, and Reasoning Process, which fixes the practical boundary between what the Neural component can propose and what the symbolic side can actually check. When that boundary is weak because the domain lacks a stable symbolic notion of correctness or the environment cannot return grounded feedback, the system may still generate candidates, but it cannot accept, reject, or repair them on principled terms. Verification falls back to heuristic scoring or manual inspection. That front-loaded burden leaves a second difficulty in the controller and interfaces that must preserve these conditions during repeated execution. Even after the selection of a suitable domain and the construction of a workable environment, the construction of a Neuro-Symbolic System remains difficult because the development of a Neuro-Symbolic system then requires multiple interacting components, including the neural, symbolic, and reasoning components as well as the integration of an overarching controller, which must then keep proposals, symbolic state, and verification procedures synchronized across the operating loop. The design and development of a Neuro-Symbolic controller itself is a difficult task too, as the controller must decide when to admit a proposal, when to commit symbolic state, when to invoke the Reasoning Process, and when a checked result can leave the system as Neuro-Symbolic System Output, and each decision has to preserve consistency with the operative world defined by the environment. Thus, developing Neuro-Symbolic systems in practice is difficult

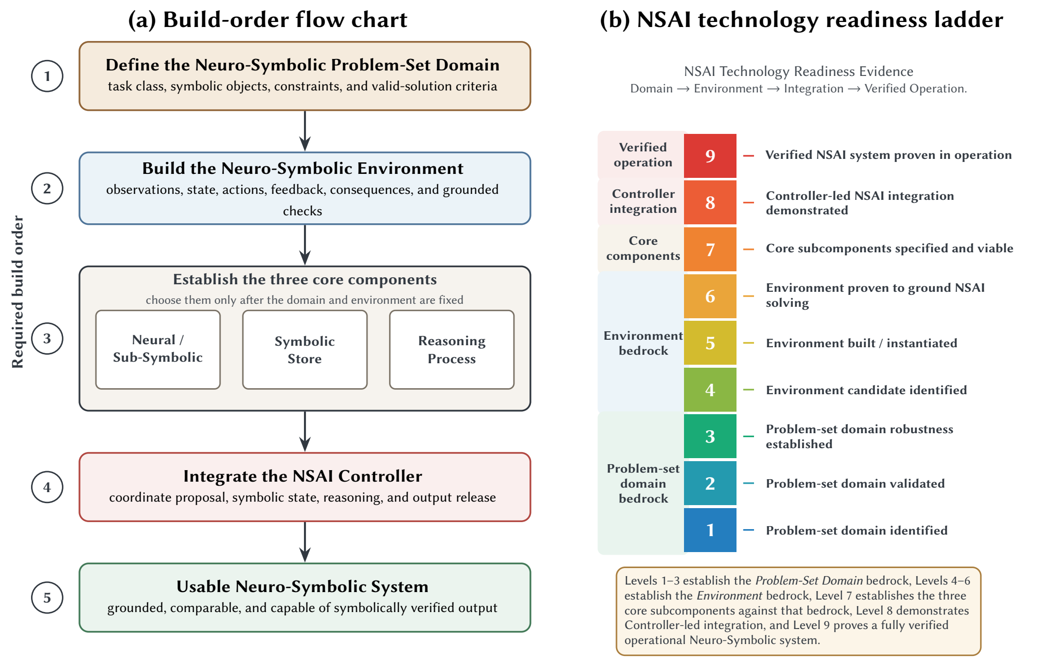

Fig. 6. **Build Order for a Neuro-Symbolic System** The figure presents the build order implied by the Neuro-Symbolic reference model. The *Neuro-Symbolic Problem-Set Domain* is the ground on which a Neuro-Symbolic system can be built, as it defines the task class, constraints, success conditions, and symbolic structure that make a problem meaningful and worth solving. From that ground one may then construct a *Neuro-Symbolic Environment* (shown as the foundation of the Neuro-Symbolic structure), which is the operative world in which a Neuro-Symbolic system can function, maintain state, reason, act, and produce consequences. Only after these two foundational layers are correctly specified can the three core pillars of the system be meaningfully established, the *Neural/Sub-Symbolic* component, the *Symbolic Store*, and the *Reasoning Process*. The *Neuro-Symbolic Controller* is then built over and across these pillars to coordinate the overall Neuro-Symbolic process as a usable system. The central point is that Neuro-Symbolic difficulty is front-loaded as a useful Neuro-Symbolic system is not created by combining neural and symbolic parts, but rather depends first on identifying the right problem-set domain and then on building an environment that can ground system operation. The order of build is thus paramount to the success of a functional Neuro-Symbolic system because if the ground is poorly chosen or the foundation is weak, the system built above it will not be functional, stable, or useful. Panel (b) recasts the same dependency as a proposed nine-level NSAI technology readiness ladder in which Levels 1–3 establish the Problem-Set Domain bedrock, Levels 4–6 establish the Environment bedrock, Level 7 establishes the three core subcomponents, Level 8 demonstrates Controller-led integration, and Level 9 proves a fully verified operational Neuro-Symbolic system.

because success depends first on selecting a domain that genuinely admits symbolically checkable problems and building an environment that grounds those checks, and then on engineering a tightly coordinated set of neural, symbolic, reasoning, and controller components whose interactions preserve correctness throughout the operating loop.

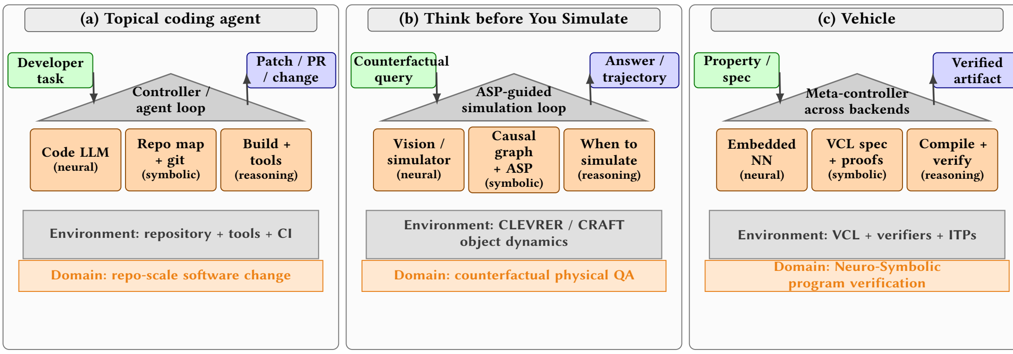

Fig. 7. **Worked cases showing how the Neuro-Symbolic foundations logic instantiates in practice.** Each panel reuses the same simplified foundations logic as the Neuro-Symbolic foundations figure. In every case, a tasking actor supplies the objective, the *problem-set domain* defines the class of solvable problems, the *environment* provides the operative world, and a controller coordinates three active pillars, which are a neural component, a symbolic store, and a reasoning process. Panel (a) is a topical repository coding agent, where a code LLM operates over repository state and executable tooling. Panel (b) abstracts *Think before You Simulate*, where symbolic ASP control decides when neural simulation is needed for counterfactual physical question answering. Panel (c) abstracts *Vehicle: Bridging the Embedding Gap in the Verification of Neuro-Symbolic Programs*, where higher-level orchestration coordinates specifications, verifier backends, and proof export. The three cases show that useful Neuro-Symbolic systems are grounded in explicit environments and orchestrated symbolic structure.

7.2 Neuro-Symbolic Problem-Set Domains

A usable Neuro-Symbolic Problem-Set Domain must specify a symbolically verifiable target artifact, the operations and constraints defined over that artifact, and the success condition that decides validity before any inner components of a Neuro-Symbolic System can be designed. Domains or problem-sets that cannot supply symbolically verifiable target artifacts are poor fits for Neuro-Symbolic systems, because the Neural component may still generate plausible candidates, but the symbolic side has no stable basis on which to accept, reject, or repair them. Figure 6 places the Problem-Set Domain at the ground of the build order for this reason, since task class, symbolic structure, and correctness conditions must exist before the Environment, Symbolic Store, Reasoning Process, and controller can be specified. All 85 of the Neuro-Symbolic systems from the verified corpus are situated in an effective Neuro-Symbolic Problem-Set Domain. LINC and CodePlan are two effective examples of Neuro-Symbolic systems that demonstrate a domain with a symbolically verifiable target artifact. LINC utilizes first-order logic formulas as the candidate artifact and theorem proving as the acceptance criterion [70], while CodePlan fixes the object of success at repository scope and ties validity to dependency-aware planning and oracle checks over repository state [11]. When a problem-set leaves that symbolic target or its scope of validity underspecified, verification collapses into heuristic scoring, string matching, or manual inspection, and the rest of the architecture loses the stable boundary that makes functional Neuro-Symbolic processes possible.

7.3 Neuro-Symbolic Environment construction

A Neuro-Symbolic Environment is the decisive condition for functional Neuro-Symbolic systems because the Neuro-Symbolic Environment is the operative world that gives symbolic reasoning something to observe, update, test, and verify. Figure 3 places the environment around the controller, Neural component, Symbolic Store, and Reasoning Process, and Figure 6 places it beneath them in the build order, which means it must expose observations, state, actions, constraints, feedback, and consequences before the inner components can function coherently. All 85 of the Neuro-Symbolic systems from the verified corpus employ an effective Neuro-Symbolic Environment. Dynamic Planning with an LLM shows the requirement for a strong foundational Neuro-Symbolic Environment in an interactive planning setting, where Alfworld, symbolic world state, beliefs, and a PDDL transition model let the system execute actions, revise beliefs, and replan against grounded consequences [21]. AlphaTrans shows the same requirement in a software setting, where repositories, ASTs, GraalVM isolation checks, and translated tests define an environment in which candidate translations succeed or fail through executable consequences [40]. When that environment is underspecified, actions become untestable, outputs become unverifiable, or the state cannot be related to consequences, so the system may still generate plausible candidates, but verification collapses into heuristic scoring or manual inspection.

7.4 Neuro-Symbolic alignment

Neuro-Symbolic alignment is the interface and semantics condition that makes symbolic verification apply to the same object that the Neural component proposes. Figure 3 demonstrates this requirement as proposals move from Neural State into Symbolic State before the Reasoning Process can apply symbolic operations, so alignment requires agreement on symbolic form, argument structure, type, and grounding. LINC demonstrates Neuro-Symbolic alignment in a formal language setting, where natural language must be translated into first-order logic that the prover can parse and interpret as the intended claim [70], while AlphaTrans shows Neuro-Symbolic alignment in a software setting, where each generated fragment must fit a static analysis schema, target skeleton, and type map before tests and runtime checks can reason over repository behavior [40].

7.5 Verification

Verification in a Neuro-Symbolic System is the mechanism that converts a neural proposal into accepted Neuro-Symbolic System Output by checking it against explicit symbolic conditions before release. Because the Neuro-Symbolic Problem-Set Domain fixes what counts as a valid symbolic object and the Neuro-Symbolic Environment supplies the operative checks that can confirm or reject that object, verification is the process that ensures system outputs are correct with respect to the stated specification, limiting the requirement for continuous human oversight. Figure 3 places the Reasoning Process between the symbolic state and Neuro-Symbolic System Output because a candidate becomes an output only after the symbolic side has accepted it, rejected it, or returned it for repair. Autoformalizing Euclidean Geometry shows the verification role under a formal proof regime, where generated theorems and proofs are accepted only after theorem equivalence and Lean proof checking [69]. LaM4Inv also demonstrates the verification role under program verification, where bounded model checking filters candidate predicates and SMT checking confirms whether the reassembled invariant satisfies the required conditions before release [84]. Without such verification, a Neuro-Symbolic System still produces proposals, but it no longer provides the symbolic acceptance trace that distinguishes a correct result from a plausible guess.

7.6 A topical example: LLM coding agents are, by nature, Neuro-Symbolic systems

LLM coding agents, by nature, fit the Neuro-Symbolic Reference Model because repository-scale software change already provides a symbolically specified Problem-Set Domain and an operative environment in which neural proposals can be stored, reasoned over, and verified. In Figure 3, the developer task becomes Neuro-Symbolic System Input provided by a human orchestrator, the code LLM is the Neural component, repository state, dependency relations, git history, file schemas, and failure traces form the Symbolic Store, and lastly the system's build and compilation process, static analyzers, and code tests act as the Reasoning Process. Finally, the agent scaffold serves as the Neuro-Symbolic Controller that schedules proposal, execution, and repair. Current products exhibit this same structure, since Claude Code is documented as an agentic coding tool that reads a codebase, edits files, runs commands, and integrates with development tools[1], and similarly Codex is documented as a coding agent that navigates a repository, edits files, runs commands, and executes tests from a prompt or specification[2]. Open-source systems that follow the same setup, such as CodePlan provides a more in-depth description of this Neuro-Symbolic setup as CodePlan instantiates the same mapping inside the verified corpus through dependency-aware planning and repository-level oracle validation [11], and Neuro-Symbolic Repair of Test Flakiness does so through localization, patch generation, recompilation, rerun testing, and feedback-driven repair [17]. The practical consequence is that coding agents succeed where many other agent settings remain unstable because code lives in a symbolic domain with explicit artifacts, executable semantics, and fast verification, so the controller can reject locally plausible patches that break compilation, tests, or repository state instead of requiring trust in the neural model. Coding agents have succeeded where other agent settings remain unstable because software engineering already provided the two outer layers that every functional Neuro-Symbolic system requires before its inner components can operate, a Problem-Set Domain with explicit symbolic correctness conditions in the form of compilation, tests, and static analysis, and an Environment in the form of mature development toolchains, version control, and executable feedback, meaning that the hard front-loaded design work was already done, and the neural component simply had to be connected to infrastructure that had been built to verify code for decades.

8 Conclusion

Neuro-Symbolic systems become functionally useful only when they can turn neural proposals into accepted Neuro-Symbolic System Output through explicit symbolic verification. The proposed reference model for Neuro-Symbolic systems was derived from 85 independently rebuilt and rerun systems. Across the verified corpus, the same core structure recurred. A Neuro-Symbolic Problem-Set Domain fixes what counts as a valid symbolic object, a Neuro-Symbolic Environment supplies the operative checks that can confirm or reject that object, and a controller coordinates a neural component, an explicit symbolic store, and a reasoning process so that candidate outputs are verified before release. Under those conditions, the Neuro-Symbolic system can produce outputs that are correct with respect to its stated specification without requiring continuous human oversight. A useful Neuro-Symbolic system does not begin with a loose coupling of neural and symbolic modules. A useful Neuro-Symbolic system begins with a Problem-Set Domain that makes validity symbolically expressible and with an environment that can ground, test, and verify system outputs. Only after those two outer conditions are met can the inner components operate as a coherent Neuro-Symbolic system whose outputs can be verified symbolically and therefore trusted without the need for continuous human or domain-expert oversight.

8.1 Limitations

The corpus of Neuro-Symbolic systems used to develop the Neuro-Symbolic Reference Model is restricted to systems with publicly released code, which likely excludes proprietary and industrial Neuro-Symbolic systems that may exhibit different structural patterns. Additionally, the Neuro-Symbolic Reference Model is derived from the current verified corpus and may require revision as

[1]Claude Code
[2]Codex

higher-level systems, particularly Level 7 self-improving systems, emerge in the literature. Lastly, the build-order argument is grounded empirically in what worked across 85 verified systems, but has not been prospectively tested against new system development.

References

[1] Hanna Abi Akl. 2024. NeSy is alive and well: A LLM-driven symbolic approach for better code comment data generation and classification. In *Joint Proceedings of the ESWC 2024 Workshops and Tutorials (CEUR Workshop Proceedings, Vol. 3749)*. CEUR-WS.org. https://ceur-ws.org/Vol-3749/genesy-01.pdf Presented at the First International Workshop on Generative Neuro-Symbolic AI (GeNeSy 2024), co-located with the 21st European Semantic Web Conference (ESWC 2024), Hersonissos, Greece.

[2] Kamal Acharya, Mehul Lad, Liang Sun, and Houbing Song. 2025. Neurosymbolic AI for Travel Demand Prediction: Integrating Decision Tree Rules into Neural Networks. doi:10.48550/arxiv.2502.01680

[3] Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. 2022. Semantic probabilistic layers for neuro-symbolic learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 2171, 16 pages.

[4] Kareem Ahmed, Eric Wang, Kai-Wei Chang, and Guy Van den Broeck. 2022. Neuro-symbolic entropy regularization. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence (Proceedings of Machine Learning Research, Vol. 180)*. PMLR, 43–53. https://proceedings.mlr.press/v180/ahmed22a.html

[5] Mohammad Mahmudul Alam, Alexander Oberle, Edward Raff, Stella Biderman, Tim Oates, and James Holt. 2024. A walsh hadamard derived linear vector symbolic architecture. In *Proceedings of the 38th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '24)*. Curran Associates Inc., Red Hook, NY, USA, Article 89, 23 pages.

[6] Uri Alon, Frank Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. 2022. Neuro-Symbolic Language Modeling with Automaton-augmented Retrieval. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 468–485. https://proceedings.mlr.press/v162/alon22a.html

[7] Yoichi Aoki, Keito Kudo, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi, and Kentaro Inui. 2023. Empirical Investigation of Neural Symbolic Reasoning Strategies. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, 1154–1162. doi:10.18653/v1/2023.findings-eacl.86

[8] Masataro Asai and Christian Muise. 2021. Learning neural-symbolic descriptive planning models via cube-space priors: the voyage home (to STRIPS). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Yokohama, Yokohama, Japan) *(IJCAI'20)*. Article 371, 7 pages.

[9] Yaniv Aspis, Krysia Broda, Jorge Lobo, and Alessandra Russo. 2022. Embed2Sym - Scalable Neuro-Symbolic Reasoning via Clustered Embeddings. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*. 421–431. doi:10.24963/kr.2022/44

[10] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic Tensor Networks. *Artificial Intelligence* 303 (2022), 103649. doi:10.1016/j.artint.2021.103649

[11] Ramakrishna Bairi, Atharv Sonwane, Aditya Kanade, Vageesh D. C., Arun Iyer, Suresh Parthasarathy, Sriram Rajamani, B. Ashok, and Shashank Shet. 2024. CodePlan: Repository-Level Coding using LLMs and Planning. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 675–698. doi:10.1145/3643757

[12] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Mateo Espinosa Zarlenga, Lucie Charlotte Magister, Alberto Tonda, Pietro Lió, Frederic Precioso, Mateja Jamnik, and Giuseppe Marra. 2023. Interpretable neural-symbolic concept reasoning. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 76, 25 pages.

[13] Celeste Barnaby, Qiaochu Chen, Roopsha Samanta, and Işıl Dillig. 2023. ImageEye: Batch Image Processing using Program Synthesis. *Proceedings of the ACM on Programming Languages* 7, PLDI (2023), 686–711. doi:10.1145/3591248

[14] Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. 2021. Neural Symbolic Regression that Scales. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 936–945. https://proceedings.mlr.press/v139/biggio21a.html

[15] Tommaso Carraro, Alessandro Daniele, Fabio Aiolli, and Luciano Serafini. 2024. Mitigating data sparsity via neuro-symbolic knowledge transfer. In *Lecture Notes in Computer Science*. Springer Nature Switzerland, Cham, 226–242.

[16] David Chanin and Anthony Hunter. 2023. Neuro-symbolic Commonsense Social Reasoning. doi:10.48550/arxiv.2303.08264

[17] Yang Chen and Reyhaneh Jabbarvand. 2024. Neurosymbolic Repair of Test Flakiness. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*. ACM, 1402–1414. doi:10.1145/3650212.3680369

[18] Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. In *Proceedings of the*

2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 6279–6292. doi:10.18653/v1/2022.emnlp-main.421
[19] Brandon Colelough, Vladimir Martirosyan, Ishan Tamrakar, William Regli, Aditya Kumar, Anh N. Nhu, Dhruv Dubey, Raj Ambavane, and Haowei Deng. 2025. Reproducibility in Computer Science: An Empirical Multi-Year Replication Study of Neuro-Symbolic AI. Manuscript under review. Under review at the International Conference on Software Maintenance and Evolution (ICSME), RENE track.
[20] Brandon C Colelough and William Regli. 2025. Neuro-Symbolic AI in 2024: A Systematic Review. (Jan. 2025). arXiv:2501.05435 [cs.AI]
[21] Gautier Dagan, Frank Keller, and Alex Lascarides Keller. 2024. Dynamic Planning with a LLM. In *Proceedings of the Language Gamification Workshop 2024 at NeurIPS*. Neural Information Processing Systems Foundation (NeurIPS), Vancouver, Canada, 1–14. doi:10.48550/arXiv.2308.06391 Presented at the Language Gamification Workshop, NeurIPS 2024.
[22] Matthew L. Daggitt, Wen Kokke, Robert Atkey, Ekaterina Komendantskaya, Natalia Slusarz, and Luca Arnaboldi. 2025. Vehicle: Bridging the Embedding Gap in the Verification of Neuro-Symbolic Programs. In *10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 337)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2:1–2:20. doi:10.4230/LIPIcs.FSCD.2025.2
[23] Alessandro Daniele and Luciano Serafini. 2022. Knowledge Enhanced Neural Networks for relational domains. doi:10.48550/arxiv.2205.15762
[24] Lauren Nicole DeLong, Yojana Gadiya, Paola Galdi, Jacques D. Fleuriot, and Daniel Domingo-Fernández. 2024. MARS: A neurosymbolic approach for interpretable drug discovery. doi:10.48550/arxiv.2410.05289
[25] Varun Dhanraj and Chris Eliasmith. 2025. Improving Rule-based Reasoning in LLMs using Neurosymbolic Representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 30589–30608. doi:10.18653/v1/2025.emnlp-main.1556
[26] Charles Dickens, Changyu Gao, Connor Pryor, Stephen Wright, and Lise Getoor. 2024. Convex and bilevel optimization for neural-symbolic inference and learning. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 432, 32 pages.
[27] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2019. Neural Logic Machines. In *International Conference on Learning Representations*. https://iclr.cc/virtual/2019/poster/816 Poster presentation at the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, Louisiana, United States.
[28] Yijiang Dong, Lara Martin, and Chris Callison-Burch. 2023. CORRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 13152–13168. doi:10.18653/v1/2023.findings-acl.832
[29] Thomas Eiter, Nelson Nicolas Higuera Ruiz, and Johannes Oetsch. 2023. A Modular Neurosymbolic Approach for Visual Graph Question Answering. In *Proceedings of the 17th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy 2023) (CEUR Workshop Proceedings, Vol. 3432)*. CEUR-WS.org, 139–149. doi:10.34726/5409
[30] Mark Endo, Joy Hsu, Jiaman Li, and Jiajun Wu. 2023. Motion question answering via modular motion programs. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 373, 17 pages.
[31] Richard Evans, Matko Bošnjak, Lars Buesing, Kevin Ellis, David Pfau, Pushmeet Kohli, and Marek Sergot. 2021. Making sense of raw input. *Artificial Intelligence* 299 (2021), 103521. doi:10.1016/j.artint.2021.103521
[32] J K F. 1928. Tychonis Brahe Dani opera Omnia. *Nature* 122, 3082 (Nov. 1928), 803–804.
[33] Reuben Feinman and Brenden M. Lake. 2021. Learning Task-General Representations with Generative Neuro-Symbolic Modeling. In *International Conference on Learning Representations*. https://openreview.net/forum?id=qzBUIzq5XR2 Poster presentation at the 9th International Conference on Learning Representations (ICLR 2021), virtual conference.
[34] Yufei Feng, Xiaoyu Yang, Xiaodan Zhu, and Michael Greenspan. 2022. Neuro-symbolic Natural Logic with Introspective Revision for Natural Language Inference. *Transactions of the Association for Computational Linguistics* 10 (2022), 240–256. doi:10.1162/tacl_a_00458
[35] Claire Glanois, Zhaohui Jiang, Xuening Feng, Paul Weng, Matthieu Zimmer, Dong Li, Wulong Liu, and Jianye Hao. 2022. Neuro-Symbolic Hierarchical Rule Induction. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 7583–7615. https://proceedings.mlr.press/v162/glanois22a.html
[36] Nanxu Gong, Wangyang Ying, Dongjie Wang, and Yanjie Fu. 2025. Neuro-Symbolic Embedding for Short and Effective Feature Selection via Autoregressive Generation. *ACM Trans. Intell. Syst. Technol.* 16, 2, Article 37 (Feb. 2025), 21 pages. doi:10.1145/3709011
[37] Safayat Bin Hakim, Muhammad Adil, Alvaro Velasquez, and Houbing Herbert Song. 2025. ANSR-DT: An Adaptive Neuro-Symbolic Learning and Reasoning Framework for Digital Twins. doi:10.48550/arxiv.2501.08561

[38] Phillip Howard, Junlin Wang, Vasudev Lal, Gadi Singer, Yejin Choi, and Swabha Swayamdipta. 2024. NeuroComparatives: Neuro-Symbolic Distillation of Comparative Knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, 4502–4520. doi:10.18653/v1/2024.findings-naacl.281

[39] Xiang Hu, Xinyu Kong, and Kewei Tu. 2023. A Multi-Grained Self-Interpretable Symbolic-Neural Model For Single/Multi-Labeled Text Classification. In *International Conference on Learning Representations*. https://openreview.net/forum?id=MLJ5TF5FtXH In-person poster presentation at the 11th International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda.

[40] Ali Reza Ibrahimzada, Kaiyao Ke, Mrigank Pawagi, Muhammad Salman Abid, Rangeet Pan, Saurabh Sinha, and Reyhaneh Jabbarvand. 2025. AlphaTrans: A Neuro-Symbolic Compositional Approach for Repository-Level Code Translation and Validation. *Proceedings of the ACM on Software Engineering* 2, FSE (2025), 2454–2476. doi:10.1145/3729379

[41] Adam Ishay, Zhun Yang, and Joohyung Lee. 2023. Leveraging large language models to generate answer set programs. In *Proceedings of the Twentieth International Conference on Principles of Knowledge Representation and Reasoning*. International Joint Conferences on Artificial Intelligence Organization, 374–383. doi:10.24963/kr.2023/37

[42] Adam Ishay, Zhun Yang, Joohyung Lee, Ilgu Kang, and Dongjae Lim. 2024. Think before You Simulate: Symbolic Reasoning to Orchestrate Neural Computation for Counterfactual Question Answering. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 6684–6693. doi:10.1109/wacv57701.2024.00656

[43] Xin Kang, Veronika Shteyngardt, Yuhan Wang, and Dov Dori. 2025. Neuro-Conceptual Artificial Intelligence: Integrating OPM with Deep Learning to Enhance Question Answering Quality. In *Proceedings of Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ COLING 2025*, Kang Liu, Yangqiu Song, Zhen Han, Rafet Sifa, Shizhu He, and Yunfei Long (Eds.). ELRA and ICCL, Abu Dhabi, UAE, 71–85. https://aclanthology.org/2025.neusymbridge-1.8/

[44] Erkan Karabulut, Paul Groth, and Victoria Degeler. 2025. Neurosymbolic Association Rule Mining from Tabular Data. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (Proceedings of Machine Learning Research, Vol. 284)*, Leilani H. Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken (Eds.). PMLR, 565–588. https://proceedings.mlr.press/v284/karabulut25a.html

[45] Garrett E. Katz, Akshay, Gregory P. Davis, Rodolphe J. Gentili, and James A. Reggia. 2021. Tunable Neural Encoding of a Symbolic Robotic Manipulation Algorithm. *Frontiers in Neurorobotics* 15 (2021). doi:10.3389/fnbot.2021.744031

[46] Jack Kelly, Alex Calderwood, Noah Wardrip-Fruin, and Michael Mateas. 2023. There and back again: extracting formal domains for controllable neurosymbolic story authoring. In *Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Salt Lake City) *(AIIDE '23)*. AAAI Press, Article 7, 11 pages. doi:10.1609/aiide.v19i1.27502

[47] Johannes Kepler. 1609. *Astronomia nova aitiologetos [romanized]*. G. Voegelinus. doi:10.5479/sil.126675.39088002685477

[48] Simon Kohaut, Felix Divo, Benedict Flade, Devendra Singh Dhami, Julian Eggert, and Kristian Kersting. 2025. The Constitutional Filter: Bayesian Estimation of Compliant Agents. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3092–3099. doi:10.1109/IROS60139.2025.11246215

[49] Simon Kohaut, Benedict Flade, Daniel Ochs, Devendra Singh Dhami, Julian Eggert, and Kristian Kersting. 2025. Probabilistic Mission Design for Neuro-Symbolic Unmanned Aircraft Systems. *IEEE Transactions on Intelligent Transportation Systems* 26, 12 (2025), 22751–22760. doi:10.1109/tits.2025.3609835

[50] Panagiotis Kouvaros, Elena Botoeva, and Cosmo De Bonis-Campbell. 2024. Formal Verification of Parameterised Neural-symbolic Multi-agent Systems. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 103–110. doi:10.24963/ijcai.2024/12 Main Track.

[51] Joshua Shay Kricheli, Khoa Vo, Aniruddha Datta, Spencer Ozgur, and Paulo Shakarian. 2024. Error Detection and Constraint Recovery in Hierarchical Multi-Label Classification without Prior Knowledge. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. ACM, 3842–3846. doi:10.1145/3627673.3679918

[52] Maxat Kulmanov, Francisco J. Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T. Arold, and Robert Hoehndorf. 2024. Protein function prediction as approximate semantic entailment. *Nature Machine Intelligence* 6, 2 (2024), 220–228. doi:10.1038/s42256-024-00795-w

[53] Maxat Kulmanov, Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. 2021. Semantic similarity and machine learning with ontologies. *Brief. Bioinform.* 22, 4 (July 2021), bbaa199.

[54] Zhicheng Lee, Zhidian Huang, Zijun Yao, Jinxin Liu, Amy Xin, Lei Hou, and Juanzi Li. 2024. DiaKoP: Dialogue-based Knowledge-oriented Programming for Neural-symbolic Knowledge Base Question Answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. ACM, 5234–5238. doi:10.1145/3627673.3679229

[55] Bowen Li, Zhaoyu Li, Qiwei Du, Jinqi Luo, Wenshan Wang, Yaqi Xie, Simon Stepputtis, Chen Wang, Katia P. Sycara, Pradeep Kumar Ravikumar, Alexander Gray, Xujie Si, and Sebastian Scherer. 2024. LogiCity: Advancing Neuro-Symbolic AI with Abstract Urban Simulation. In *Advances in Neural Information Processing Systems*, Vol. 37. 69840–69864. https://proceedings.neurips.cc/paper_files/paper/2024/hash/8196be81e68289d7a9ece21ed7f5750a-Abstract-Datasets_and_Benchmarks_Track.html Poster presentation at the 38th Conference on Neural Information Processing Systems (NeurIPS 2024), Vancouver, Canada; Datasets and Benchmarks Track.

[56] Danyang Li, Mingyu Cai, Cristian-Ioan Vasile, and Roberto Tron. 2023. Learning Signal Temporal Logic through Neural Network for Interpretable Classification. In *2023 American Control Conference (ACC)*. IEEE, 1907–1914. doi:10.23919/acc55779.2023.10156357

[57] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. 2020. Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 5884–5894. https://proceedings.mlr.press/v119/li20f.html

[58] Xiao Li, Gong Cheng, Ziheng Chen, Yawei Sun, and Yuzhong Qu. 2022. AdaLoGN: Adaptive Logic Graph Network for Reasoning-Based Machine Reading Comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 7147–7161. doi:10.18653/v1/2022.acl-long.494

[59] Ziyang Li, Jiani Huang, Jason Liu, Felix Zhu, Eric Zhao, William Dodds, Neelay Velingker, Rajeev Alur, and Mayur Naik. 2024. Relational programming with foundation models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 9 (2024), 10635–10644. doi:10.1609/aaai.v38i9.28934

[60] Ziyang Li, Jiani Huang, and Mayur Naik. 2023. Scallop: A Language for Neurosymbolic Programming. *Proceedings of the ACM on Programming Languages* 7, PLDI (2023), 1463–1487. doi:10.1145/3591280

[61] Zenan Li, Zhaoyu Li, Wen Tang, Xian Zhang, Yuan Yao, Xujie Si, Fan Yang, Kaiyu Yang, and Xiaoxing Ma. 2025. Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=FiyS0ecSm0 Poster presentation at the 13th International Conference on Learning Representations (ICLR 2025), Singapore.

[62] Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven Data Construction for Zero-shot Evaluation in Commonsense Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 15 (2021), 13507–13515. doi:10.1609/aaai.v35i15.17593

[63] Jaron Maene, Vincent Derkinderen, and Pedro Zuidberg Dos Martires. 2024. KLay: Accelerating Arithmetic Circuits for Neurosymbolic AI. In *International Conference on Learning Representations*. https://api.semanticscholar.org/CorpusID:273350969

[64] Nikolaos Manginas, George Paliouras, and Luc De Raedt. 2025. NeSyA: Neurosymbolic Automata. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 5950–5958. doi:10.24963/ijcai.2025/662

[65] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. 2024. bears make neuro-symbolic models aware of their reasoning shortcuts. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence* (Barcelona, Spain) *(UAI '24)*. JMLR.org, Article 114, 35 pages.

[66] Mohamed Mejri, Chandramouli Amarnath, and Abhijit Chatterjee. 2024. LARS-VSA: A Vector Symbolic Architecture For Learning with Abstract Rules. doi:10.48550/arxiv.2405.14436

[67] Eleonora Misino, Giuseppe Marra, and Emanuele Sansone. 2022. VAEL: bridging variational autoencoders and probabilistic logic programming. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 337, 13 pages.

[68] Adithya Murali, Atharva Sehgal, Paul Krogmeier, and P Madhusudan. 2022. Composing neural learning and symbolic reasoning with an application to visual discrimination. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence* (Vienna, Austria). International Joint Conferences on Artificial Intelligence Organization, California.

[69] Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. 2024. Autoformalizing euclidean geometry. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 1497, 47 pages.

[70] Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5153–5176. doi:10.18653/v1/2023.emnlp-main.313

[71] Claire Benet Post, Marie C. McGregor, Maria Leonor Pacheco, and Alexis Palmer. 2024. Accelerating UMR adoption: Neuro-symbolic conversion from AMR-to-UMR with low supervision. In *Proceedings of the Fifth International Workshop*

on Designing Meaning Representations @ LREC-COLING 2024. ELRA, 140–150. doi:10.63317/28rm2npp63im

[72] Tanawan Premsri and Parisa Kordjamshidi. 2025. Neuro-symbolic Training for Reasoning over Spatial Language. In *Findings of the Association for Computational Linguistics: NAACL 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 2395–2414. doi:10.18653/v1/2025.findings-naacl.128

[73] Henrijs Princis, Cristina David, and Alan Mycroft. 2025. Enhancing SQL query generation with neurosymbolic reasoning. *Proc. Conf. AAAI Artif. Intell.* 39, 19 (April 2025), 19959–19968.

[74] Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Yang Wang, and Lise Getoor. 2023. NeuPSL: Neural Probabilistic Soft Logic. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 4145–4153. doi:10.24963/ijcai.2023/461

[75] Xin Quan, Marco Valentino, Danilo Carvalho, Dhairya Dalal, and Andre Freitas. 2025. PEIRCE: Unifying Material and Formal Reasoning via LLM-Driven Neuro-Symbolic Refinement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, 11–21. doi:10.18653/v1/2025.acl-demo.2

[76] Marc Roig Vilamala, Tianwei Xing, Harrison Taylor, Luis Garcia, Mani Srivastava, Lance Kaplan, Alun Preece, Angelika Kimmig, and Federico Cerutti. 2023. DeepProbCEP: A neuro-symbolic approach for complex event processing in adversarial settings. *Expert Systems with Applications* 215 (2023), 119376. doi:10.1016/j.eswa.2022.119376

[77] Arseny Skryagin, Daniel Ochs, Devendra Singh Dhami, and Kristian Kersting. 2023. Scalable Neural-Probabilistic Answer Set Programming. *Journal of Artificial Intelligence Research* 78 (2023), 579–617. doi:10.1613/jair.1.15027

[78] Tanel Tammet, Priit Järv, Martin Verrev, and Dirk Draheim. 2023. An Experimental Pipeline for Automated Reasoning in Natural Language (Short Paper). In *Automated Deduction – CADE 29: 29th International Conference on Automated Deduction, Rome, Italy, July 1–4, 2023, Proceedings* (Rome, Italy). Springer-Verlag, Berlin, Heidelberg, 509–521. doi:10.1007/978-3-031-38499-8_29

[79] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature* 625, 7995 (Jan. 2024), 476–482.

[80] Ho Fung Tsoi, Vladimir Loncar, Sridhara Dasu, and Philip Harris. 2025. SymbolNet: Neural symbolic regression with adaptive dynamic pruning for compression. *Machine Learning: Science and Technology* 6, 1 (2025), 015021. doi:10.1088/2632-2153/adaad8

[81] Emile van Krieken, Pasquale Minervini, Edoardo Maria Ponti, and Antonio Vergari. 2025. Neurosymbolic Diffusion Models. In *Advances in Neural Information Processing Systems*. 1–43. doi:10.48550/arXiv.2505.13138 Poster presentation at the 39th Annual Conference on Neural Information Processing Systems (NeurIPS 2025), San Diego, California, United States.

[82] Emile van Krieken, Thiviyan Thanapalasingam, Jakub M. Tomczak, Frank van Harmelen, and Annette ten Teije. 2023. A-NeSI: A Scalable Approximate Method for Probabilistic Neurosymbolic Inference. In *Advances in Neural Information Processing Systems*, Vol. 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/4d9944ab3330fe6af8efb9260aa9f307-Abstract-Conference.html Poster presentation at the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, Louisiana, United States.

[83] Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. 2022. DeepStochLog: Neural Stochastic Logic Programming. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 9 (2022), 10090–10100. doi:10.1609/aaai.v36i9.21248

[84] Guangyuan Wu, Weining Cao, Yuan Yao, Hengfeng Wei, Taolue Chen, and Xiaoxing Ma. 2024. LLMMeetsBoundedModelChecking: Neuro-symbolic Loop Invariant Inference. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*. ACM, 406–417. doi:10.1145/3691620.3695014

[85] Yuxuan Wu and Hideki Nakayama. 2024. MILE: Memory-Interactive Learning Engine for Neuro-Symbolic Solutions to Mathematical Problems. *IEEE Access* 12 (2024), 134355–134365. doi:10.1109/access.2024.3461150

[86] Zhun Yang, Adam Ishay, and Joohyung Lee. 2023. Learning to Solve Constraint Satisfaction Problems with Recurrent Transformer. In *International Conference on Learning Representations*. https://iclr.cc/virtual/2023/poster/11188 Poster presentation at the 11th International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda.

[87] Zhun Yang, Joohyung Lee, and Chiyoun Park. 2022. Injecting Logical Constraints into Neural Networks via Straight-Through Estimators. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 25096–25122. https://proceedings.mlr.press/v162/yang22h.html

[88] Congchi Yin, Feng Li, Shu Zhang, Zike Wang, Jun Shao, Piji Li, Jianhua Chen, and Xun Jiang. 2025. MDD-5k: a new diagnostic conversation dataset for mental disorders synthesized via neuro-symbolic LLM agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 24 (2025), 25715–25723. doi:10.1609/aaai.v39i24.34763

[89] Jie Ying, Haowei Lin, Chao Yue, Yajie Chen, Chao Xiao, Quanqi Shi, Yitao Liang, Shing-Tung Yau, Yuan Zhou, and Jianzhu Ma. 2025. A neural symbolic model for space physics. *Nat. Mach. Intell.* 7, 10 (Oct. 2025), 1726–1741.

Appendix A. Verified corpus entries used in the comparative synthesis

Editorial note. This revision replaces the prior tiered split with one consolidated table covering all 85 independently rerun systems. Architecture family records the dominant coupling motif rather than a separate architecture class. The revised family labels are: lifting pipeline, constraint-shaped learner, joint-inference solver, proposal-verify-repair, situated orchestration, artifact synthesis / induction, and boundary / adjunct. Neuro-Symbolic Level reports the primary level assignment from the reconciled master comparative synthesis. Orchestrator / Coupling now records the external tasking actor (where identifiable) plus the dominant neural-symbolic coupling pattern, rather than defaulting to a generic runtime label.

Abbreviations. Where a paper does not expose a distinct external orchestrator, the Orchestrator / Coupling column names the most defensible tasking actor or workflow context together with the dominant coupling pattern. Family labels are dominant motifs, not mutually exclusive bins. Some entries instantiate one or more required roles as no-op implementations. In this table, no-op status is always stated at the role level, for example no-op controller, no-op runtime reasoner, no-op symbolic runtime, or no-op meta-controller. A no-op role is explicitly present in the reproduced workflow but returns a pass-through, identity, empty, default, or fixed result without materially transforming system state in the evaluated configuration. No-op is therefore different from absent: absent roles do not satisfy the functional Neuro-Symbolic structure, while no-op roles preserve the interface contract in lower-boundary systems.

Table A1. All 85 verified entries: internal component mapping, orchestration, architecture family, and Neuro-Symbolic level.

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Neural-to-symbolic lifting pipelines									
AMR-to-UMR [71]	DMR @ LREC-COLING / 2024	Animacy models; Sentence-BERT; FFNN	Role rules from UMR guidelines	Rule-guided classification	Fixed modular pipeline	Corpus task harness; staged neural+rule conversion coupling	Lifting pipeline	Level 2	Low-data hybrid focused on split-role mapping rather than full conversion.
Embed2Sym [9]	KR special session / 2022	Perception embeddings + neural reasoner	ASP rules; cluster-label mapping	ASP optimization and inference	Staged lifting pipeline	Task harness; staged lift-to-symbol + ASP runtime coupling	Lifting pipeline	Level 1	Strong scalability story; codebase and venue-year details remain uncertain.
Rule-based LLM reasoning [25]	EMNLP / 2025	Base LLM plus learned encoder/decoder modules	HRR/VSA vectors, problem-type tags, and exact rule-based algorithms	Execution of deterministic symbolic algorithms in VSA space	Fixed intervention pipeline with similarity-gated symbolic execution	User prompt at inference; fixed encode-exact-rule-decode coupling	Lifting pipeline	Level 1	Strong neural-symbolic interface, but the conservative level reading is still pipeline-like.
LINC [70]	EMNLP / 2023	LLM semantic parser	FOL formulas; Prover9	Theorem proving	Parse to prove to vote	User / benchmark tasking; semantic-parse-to-theorem-prover coupling	Lifting pipeline	Level 1	Formal prover is central; interaction is mostly one-way.
LLM-to-ASP [41]	KR (Main Track) / 2023	GPT-3 / GPT-4	ASP programs, constraints	ASP solving	Fixed prompt pipeline	User / benchmark puzzle prompt; NL-to-ASP-to-solver coupling	Lifting pipeline	Level 1	Canonical neural-to-symbolic pipeline.
NCAI [43]	Neusym-Bridge Workshop @ COLING / 2025	LLM for NL to OPM and QA	OPM / OPD / OPL / objects / processes / states	LLM-mediated reasoning over OPM knowledge	Explicit 2-stage pipeline	User question / process text; two-stage NL-to-OPM-to-QA coupling	Lifting pipeline	Level 0	Symbolic structure is explicit, but a separate symbolic inference engine is not described.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Commonsense social reasoning [16]	arXiv / 2023	AMR parsing support plus contextual RoBERTa embeddings used in similarity-based unification and merged-node representations.	AMR trees, merged AMR variants, first-order implications for ROTs, grounded facts for SSTs, and theorem-prover clauses.	Resolution-based theorem proving with non-binary unification driven partly by embedding similarity.	An explicit but simple pipeline: parse, generate merged variants, convert to logic, and query the theorem prover.	Sentence-pair tasking; parse-to-logic-to-theorem-prover coupling	Lifting pipeline	Level 1	The system has a genuine logic-and-proof core, but its task scope is narrow and the source reporting emphasizes trend curves more than single headline scores.
NSGRAPH VGQA [29]	NeSy Workshop / 2023	OCR + graph recognition	Graph facts; question program; ASP theory	ASP graph reasoning	Fixed modular pipeline	Graph-QA benchmark harness; neural parsing-to-ASP-reasoning coupling	Lifting pipeline	Level 1	Clean pipeline case: perceptual front-end followed by symbolic QA.
Symbolic-Neural text classifier [39]	ICLR / 2023	Structured LM; MLP; top-down encoder	Parse trees; label trees; yield function; DP constraints	Dynamic programming over trees	Parse -> node scores -> yield	Benchmark harness; neural parse-to-symbolic-tree extraction coupling	Lifting pipeline	Level 1	Symbolic structure is explicit but reasoning depth is modest.
Constraint-shaped differentiable learners									
CL-STE [87]	ICML / 2022	MLP/CNN/GNN outputs with STE	CNF theory; facts; clause matrix	Clause-wise deduction/satisfaction loss; not full symbolic solving	NN -> binarize -> CNF loss -> STE update	Training harness; CNF-loss-through-STE coupling	Constraint-shaped learner	Level 2	Clear example of constraint-integrated, weakly neuro-symbolic design.
DCR [12]	ICML / 2023	Concept encoder; role/relevance MLPs	Fuzzy rules over concept truths	Fuzzy rule execution	Generate rule -> execute rule	Task harness; neural rule-generation + fuzzy-rule execution coupling	Constraint-shaped learner	Level 2	Strong interpretability evidence; global behavior only partly summarized.
DeepProbCEP [76]	ESWA / 2023	Perception NN for simple events	ProbLog rules; sequence framework	Probabilistic logic inference	Windowing -> NN -> logic	Stream / benchmark harness; neural perception + differentiable logic coupling	Constraint-shaped learner	Level 2	Strong on sparse data; slower than neural baselines.
KENN relational domains [23]	Preprint (under review) / 2022	Dense neural classifier producing predicate pre-activations	Function-free first-order clauses; unary/binary predicates; learnable clause weights	Differentiable clause-satisfaction revision via KE/RKE residual updates	Feed-forward neural stage followed by one or more logical residual layers	Task harness; neural predictions + logical residual-revision coupling	Constraint-shaped learner	Level 2	Stacked-layer refinement could justify a higher reading, but evidence is strongest for Level 2.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Learning STL [56]	ACC / 2023	NN-TLI differentiable logic network	STL(1), predicates, temporal intervals	STL robustness reasoning with sound approximations	Training-time only	Training / task harness; differentiable STL execution coupling	Constraint-shaped learner	Level 2	Lower ranking is safer; tighter co-designed-reasoner reading remains plausible.
LTN [10]	AI / 2022	Predicate/function groundings	Real Logic formulas and KB	Fuzzy satisfiability, querying, refutation	Satisfiability-maximization loop	Task / formula harness; fuzzy-logic satisfiability coupling	Constraint-shaped learner	Level 2	Broad framework; large-scale evidence limited.
NeSy knowledge transfer [15]	ECIR / LNCS / 2024	Source and target MF models	LTN axioms; Likes/LikesGenre/HasGenre	Fuzzy logical regularization during training	Two-stage training pipeline	Training workflow; source-target logical-regularization coupling	Constraint-shaped learner	Level 2	Symbolic role is substantive but remains training-time.
NeSy Entropy [4]	UAI / 2022	Task-specific predictor	Logical constraints; circuits	Circuit-based constrained entropy	Training-time coupling	Task harness; circuit-constrained entropy coupling	Constraint-shaped learner	Level 2	Strong constraint-integrated method rather than a runtime reasoning loop.
Recurrent Transformer for CSPs [86]	ICLR / 2023	Recurrent Transformer; CNN visual embedding	CSP formalization; cardinality constraints; attention constraints	Neural iterative refinement; constraint-loss-guided optimization	Fixed recurrent unrolling; optional loss injection at all layers/steps	Training / benchmark harness; recurrent prediction + constraint-loss coupling	Constraint-shaped learner	Level 2	Symbolic content is real but concentrated in training-time losses rather than a separate runtime reasoner.
Spatial Logic Training [72]	Findings of NAACL / 2025	BERT; Flan-T5; LLM baselines	79 spatial rules; Q-Chains; DomiKnowS constraints	Forward chaining in training; soft-logic regularization	Training-time pipeline only	Training / benchmark harness; forward-chaining / soft-logic coupling	Constraint-shaped learner	Level 2	Symbolic rules shape training rather than inference-time reasoning.
Solver-integrated joint-inference reasoners									
A-NESI [82]	NeurIPS / 2023	Perception model; prediction model; optional explanation model	Worlds w; beliefs P; symbolic function c; symbolic pruner	Approximate inference for p(y|P) and p(w|y,P); beam search; pruning	Perceive -> infer -> optional explain/prune loop	Task harness; perception-to-belief-to-prune inference coupling	Joint-inference solver	Level 4	Explanation mode makes a higher ranking plausible, but the conservative primary fit remains Level 4.
Constitutional Filter [48]	IROS / 2025	Perception/trust features; one synthesis mentions an anchoring classifier	Probabilistic first-order Constitution; StaR Map	Probabilistic constitutional inference in Bayes update	Trust ratio modulates symbolic influence	Analyst / tracking feed; probabilistic constitutional-update coupling	Joint-inference solver	Level 2	Lower ranking is safer; trust mechanism can also be read as meta-control.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Convex+bilevel NS inference [26]	ICML / 2024	Task-specific neural predictors that parameterize energy terms	NeuPSL rules, atoms, linear constraints, and symbolic weights	Convex MAP inference, LCQP reformulation, dual BCD, bilevel optimization	Outer learning loop plus repeated lower-level inference	Task harness; outer learning loop over symbolic inference coupling	Joint-inference solver	Level 4	One of the clearest tightly integrated neuro-symbolic systems in the set.
DeepGO-SE [52]	Nature Machine Intelligence / 2024	ESM2 embeddings; MLP blocks; GAT variants	GO axioms; ELEmbeddings world models; ontology statements	Approximate semantic entailment across multiple models	Multi-model aggregation pipeline; optional PPI/GAT integration	Ontology / benchmark harness; neural predictions and axiom-entailment coupling	Joint-inference solver	Level 4	Strong ontology-grounded hybrid with explicit entailment-style aggregation.
DeepStochLog [83]	AAAI / 2022	Neural grammar rules / neural classifiers	SDCGs; Prolog goals; derivation trees; AND-OR circuits	Resolution; tabling; semiring circuit evaluation	Fixed logical inference and differentiable circuit pipeline	Task harness; neural grammar and derivation coupling	Joint-inference solver	Level 4	Alternative probabilistic semantics and tabling are central to its scalability story.
NeSyA [64]	IJCAI / 2025	CNN grounding network	Symbolic automata with logical transitions	WMC plus alpha-recursion	Temporal inference loop	Task harness; neural grounding and automaton inference coupling	Joint-inference solver	Level 4	One of the clearest symbolic-temporal cores in the set.
NESYDM [81]	NeurIPS / 2025	Masked diffusion unmasking model	Program phi; WMC structure	Iterative program-conditioned denoising	Reverse diffusion loop + majority vote	Task harness; program-conditioned diffusion / consistency coupling	Joint-inference solver	Level 3	Strong on path planning and calibration.
NeuPSL [74]	IJCAI / 2023	Neural predicates / neural classifiers	PSL rules; weighted hinge-loss potentials; linear constraints	Convex joint inference via energy minimization	Neural inference -> potential instantiation -> ADMM -> joint learning	Task harness; joint energy-based neural-and-PSL-inference coupling	Joint-inference solver	Level 4	Primary fit is solver-centric joint inference; a weaker reading emphasizes constraint-shaped energy.
Neural Logic Machines [27]	ICLR / 2019	Shared MLP rule modules	Predicate tensors; quantifiers; arities	Lifted layered deduction	Fixed multi-layer deduction; RL in control tasks	Task harness; learned lifted-deduction coupling	Joint-inference solver	Level 4	One of the most explicit reasoning architectures.

Continued on next page

Table A1 continued from previous page

Paper	**Venue / Year**	**Neural / Sub-Symbolic**	**Symbolic Store**	**Reasoning Process**	**Neuro-Symbolic Controller**	**Orchestrator / Coupling**	**Architecture family**	**Neuro-Symbolic Level**	**Notes**
NSPose [30]	ICML / 2023	2s-AGCN; learned concept/attribute/temporal operators	Motion DSL/programs	Recursive program execution over motion segments	Program structure controls inference	Program-query harness; neural grounding + program-execution coupling	Joint-inference solver	Level 4	Program structure governs inference over learned motion features; co-design is strong.
SLASH [77]	JAIR / 2023	NPPs: NN, PC, or NN+PC; YOLO/MLPs in VQA	ASP rules; stable models; KGs/SGs; query programs	Stable-model reasoning; probabilistic query aggregation; SAME pruning	NPP -> SAME -> ASP -> probability -> gradient loop	Task harness; NPP-to-ASP-to-probability/gradient coupling	Joint-inference solver	Level 4	SAME is an explicit pruning strategy, but the safer secondary reading stays below meta-control.
Scallop [60]	PLDI / 2023	Task-specific neural models (CNN, RoBERTa, DistilBERT, BiLSTM, FastRCNN, S3D)	Relations; Datalog-like rules; scene graphs; KB facts; SclRam IR; provenance tags	Recursive logical inference with negation/aggregation; probabilistic and differentiable provenance reasoning	Program-level orchestration via Scallop modules and input/output mappings	Program / task harness; differentiable Datalog coupling	Joint-inference solver	Level 4	Strong framework paper; symbolic reasoning is central.
SPL [3]	NeurIPS / 2022	Feature extractor + gating network	Constraint and probabilistic circuits	Exact probabilistic inference; constrained MAP	Integrated prediction layer	Task harness; neural parameterization within constrained-circuit coupling	Joint-inference solver	Level 4	Among the strongest and most tightly integrated neuro-symbolic papers in the set.
VAEL [67]	NeurIPS / 2022	VAE encoder/decoder; MLP to fact probs	ProbLog program; possible worlds	Success probability; evidence sampling	Encode -> ProbLog -> decode	Task harness; neural latent and ProbLog possible-world coupling	Joint-inference solver	Level 2	Transfer by swapping programs is a key result.
Proposal-verification-repair loops									
AdaLoGN [58]	ACL / 2022	RoBERTa text encoder, learned relevance scorer, and heterogeneous graph message passing with attention and subgraph-to-node interaction.	Text Logic Graph (TLG) with explicit logical relation labels and rule-based extension operations.	Rule-based graph extension plus graph message passing used for answer scoring.	Explicit iterative runtime loop: build raw TLG, propose symbolic extensions, admit relevant ones, run message passing, and rescore.	Benchmark question / option tasking; iterative rule-extension and rescoring coupling	Proposal-verify-repair	Level 3	The iterative neural-symbolic loop is clear, but the symbolic fragment is limited and the final answer is not formally proved.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
AlphaTrans [40]	Proc. ACM Software Engineering (FSE) / 2025	DeepSeek-Coder; GPT-4o	Schema, call graph, type map, skeleton, tests	Static analysis + validation loop	Strong explicit orchestration	Developer tasking + test/build feedback; staged translate-validate-reprompt coupling	Proposal-verify-repair	Level 3	LLM plus program analysis and layered validation rather than theorem-prover symbolicity.
Closed Loop NSL [57]	ICML / 2020	LeNet/perception net; Pointer Network setup	CFG grammar; parse trees; symbolic formulas/programs; domain knowledge	Grammar parsing; symbolic execution; back-search; MH-style interpretation	Explicit closed loop: propose, parse, execute, correct, retrain	Target answer + execution feedback; propose-parse-execute-correct coupling	Proposal-verify-repair	Level 3	One of the clearest iterative cooperative systems in the set.
CodePlan [11]	FSE / 2024	Code-editing LLM	Dependency graph; plan graph; edit specs	Impact analysis; adaptive planning	Explicit iterative loop	Developer seed edits + oracle diagnostics; iterative plan-edit-validate coupling	Proposal-verify-repair	Level 3	One of the strongest runtime neuro-symbolic systems in the set.
LaM4Inv [84]	ASE / 2024	LLM candidate generator	Predicates; invariants; BMC; SMT conditions; counterexamples	Predicate filtering, reassembly, SMT checking	Explicit closed loop	Benchmark specification + counterexamples; iterative invariant propose-check-revise coupling	Proposal-verify-repair	Level 3	One of the clearest iterative neuro-symbolic systems in the set.
Flaky-test repair [17]	ISSTA / 2024	GPT-4 or Magicoder patch generator	Program-analysis artifacts, failure traces, shared state, helper methods, and build/test context	Localization, compile-repair heuristics, and execution-based validation	Explicit Inspector to Prompt Generator to Tailor/Stitching to Validator feedback loop	Developer / flaky-test feedback; localize-patch-validate coupling	Proposal-verify-repair	Level 3	Conservative primary ranking; explicit controller could justify a higher control-centric reading.
NS-NLI [34]	TACL / 2022	GPT-2 local relation model	Natural logic relations; projection/composition rules; WordNet proposals	Natural logic execution; RL; introspective revision	Explicit training-time revision loop	Premise-hypothesis tasking + proof-state feedback; introspective revision coupling	Proposal-verify-repair	Level 5	Most explicit paper in the set on introspective reasoning-path revision and explanation.
PEIRCE [75]	ACL System Demonstrations / 2025	LLMs; retrievers; soft critics	SSKB; formalisations; Isabelle; Prolog	Conjecture-criticism refinement	Explicit iterative refinement controller	User problem + critique/prover feedback; iterative conjecture-critique-refine coupling	Proposal-verify-repair	Level 3	Most explicit critique-and-refinement loop; evaluation is framework-style rather than a single benchmark push.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Olympiad inequalities [61]	ICLR / 2025	LLM for rewriting tactics and goal ranking	Lemma/tactic library, symbolic heuristics, CAD/SMT solvers, Lean formalization	Iterative proof search with tactic generation, pruning, filtering, ranking, and formal checking	Explicit goal-search loop with tactic generation, symbolic filtering, and neural ranking	Formal goal + counterexample/prover feedback; tactic propose-prune-check coupling	Proposal-verify-repair	Level 3	Controller-centric reading could raise the level, but the conservative synthesis keeps the primary level at 3.
There and Back Again [46]	AIIDE / 2023	GPT-4 / GPT-3.5	PDDL domains/problems; plans	POCL planning + validation	Explicit pipeline with repair loop	Author prompt + planner feedback; iterative generate-plan-debug coupling	Proposal-verify-repair	Level 3	Strong hybrid creative-support pipeline, but planning success remains limited.
Xander [73]	AAAI / 2025	Pretrained LM; optional neural checker	Normalized SQL; symbolic PQC; BFS queue; repair rules	Best-first search; symbolic pruning; execution; repair	Explicit BFS/backtracking controller	User query + DB execution feedback; LM search-prune-repair coupling	Proposal-verify-repair	Level 3	Iterative LM generation plus symbolic pruning/repair is explicit; higher secondary rankings are less conservative.
Situated control and orchestration systems									
ANSR-DT [37]	arXiv / 2025	CNN-LSTM + PPO	Rule base; symbolic facts	Rule extraction and inference	Explicit layered adaptation loop	Operator / sensor-feedback loop; layered neural-symbolic adaptation coupling	Situated orchestration	Level 3	Architecturally ambitious, but evidence remains synthetic and rule scale is limited.
DiaKoP [54]	CIKM / 2024	Llama 3-70B; BART parser	KoPL; VisKoP; Wikidata	KB program execution + routing	Explicit dialogue policy and history tracker	Human dialogue user; controller-routed KB / history / LLM coupling	Situated orchestration	Level 1	Applied multi-source KBQA with explicit controller; repository reporting is inconsistent across the source analyses.
Dynamic Planning with a LLM [21]	Language Gamification Workshop @ NeurIPS / 2024	GPT-3.5 goal/belief generation	PDDL; world state W; beliefs B	Symbolic planning with BFS(f)	Closed planning-action loop	User goal + environment feedback; closed-loop belief/planner coupling	Situated orchestration	Level 3	Clear iterative neural-symbolic cooperation.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
MARS [24]	Under review (arXiv preprint) / 2024	RL agent with LSTM history encoder	KG metapaths; logical rules; learned rule weights	Rule-guided KG traversal and P2H weight updates	Episodic RL over graph paths with rule-weighted rewards	KG task / reward loop; rule-guided RL traversal coupling	Situated orchestration	Level 3	Tighter co-optimization reading is plausible; lower primary level chosen conservatively.
ProMis [49]	IEEE TITS / 2025	ChangeFormer; LLM code generation	HPLPs; probabilistic spatial relations; PML	Probabilistic logical inference	Modular clause pipeline	Mission designer + spatial priors; probabilistic mission-inference coupling	Situated orchestration	Level 1	Strong symbolic back-end with mostly one-way flow.
Think before You Simulate [42]	WACV / 2024	Percep-tion/simulation baselines; GPT-3.5/4 proxy simulator	ASP causal graph	ASP causal orchestration	Explicit symbolic module chooses simulation use	Counterfactual question + scene state; symbolic controller routes simulation calls	Situated orchestration	Level 6	Strongest explicit orchestration example in the set.
Tunable Neural Encoding [45]	Frontiers in Neuro-robotics / 2021	Neural Virtual Machine; RL fine-tuning	Symbolic VM; assembly program; mappings	Symbolic program execution emulated neurally	Gated VM execution plus RL training	Goal specification + simulator feedback; neural-VM / symbolic-program coupling	Situated orchestration	Level 4	Compiled symbolic program inside neural VM; runtime separation is limited.
VIEIRA [59]	AAAI / 2024	GPT; CLIP; ViLT; OWL-ViT; SAM; DSFD; diffusion models; others	Relations; rules; DSLs; ADTs; probabilistic tuples	Logical inference; DSL interpretation; soft joins; probabilistic reasoning	Explicit declarative orchestration through relational programs and foreign interfaces	User query / declarative program; relational orchestration across external tools	Situated orchestration	Level 3	Explicit orchestration could justify a higher reading; lower primary level chosen conservatively.
Symbolic artifact synthesis and theory/model induction systems									
Aerial+ [44]	NeSy / 2025	Denoising autoencoder	Association rules, itemsets	Rule extraction by thresholding	Simple train/extract pipeline	Offline miner / benchmark harness; autoencoder-to-rule-extraction coupling	Artifact synthesis / induction	Level 1	Explicit symbolic outputs, but comparatively light symbolic reasoning.
Autoformalizing Geometry [69]	ICML / 2024	GPT-4 / GPT-4V	LeanEuclid; E; SMT formulas	Equivalence checking; Lean proof checking	Generation followed by symbolic checking	Human / benchmark theorem prompt; generate-to-equivalence/proof-check coupling	Artifact synthesis / induction	Level 1	Very strong semantic verification; no full model-repair loop is documented.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Visual discrimination synthesis [68]	IJCAI / 2022	Pretrained vision models that detect objects, attributes, and relations and produce scene-graph information.	First-order scene models, FO-SL formulas, discriminator specifications, and SAT encodings for synthesis.	SAT-based symbolic synthesis and exact checking of discriminators against finite scene models.	Mostly one-way pipeline from perception to symbolic synthesis, with iterative deepening in quantifier count during symbolic search.	Benchmark puzzle tasking; perception-to-scene-model-to-formula-synthesis coupling	Artifact synthesis / induction	Level 1	A clean pipeline architecture with a strong formal interface; failures are dominated by perception and representational limits.
Cube-Space Priors [8]	IJCAI / 2020	SAE; AAE/Cube-Space AE; Back-to-Logit effect predictor	Binary latent propositions; STRIPS PRE/ADD/DEL; PDDL	Extracted STRIPS reasoning plus heuristic search planning	End-to-end learning, then symbolic extraction, then external planning	Goal image / task harness; learned latent-to-PDDL-planner coupling	Artifact synthesis / induction	Level 1	Strong pipeline paper; the STRIPS-compatible prior also shapes what the model can learn.
GNS [33]	ICLR / 2021	CNN/MLP/LSTM/attention modules	Program control flow; stroke primitives; symbolic renderer	Generative execution + posterior search	GenerateType loop; parse search	Task harness; neural generation under symbolic program control	Artifact synthesis / induction	Level 3	Quantitative on classification/LL, qualitative on some generation tasks.
ImageEye [13]	PLDI / 2023	Pretrained vision modules	Symbolic images; DSL; goal annotations; rewrite rules	Enumerative PBE synthesis; goal inference; partial eval	Explicit worklist synthesis	User demonstrations; perceptual grounding-to-symbolic-synthesis coupling	Artifact synthesis / induction	Level 1	Strong symbolic synthesis core; primary coupling remains neural perception feeding symbolic search.
Raw-input theory learning [31]	AI / 2021	Binary NN / pretrained recognizer	Datalog-like theory; ASP; constraints	SAT/ASP theory synthesis	Template search + joint solve	Benchmark / task harness; perception-to-theory-synthesis coupling	Artifact synthesis / induction	Level 4	Strongest explicit theory-synthesis paper in set.
MILE [85]	IEEE Access / 2024	Encoder + memory-interactive decoder	Segmented formulas; constants/numbers/references	Recurrent formula construction + symbolic execution	Explicit decoder/executor loop; no-op meta-controller above the fixed generation-execution cycle	Problem prompt + executor feedback; formula-generation / symbolic-execution coupling	Artifact synthesis / induction	Level 1	Strong symbolic execution pipeline with learning-aware structure; only the meta-controller role is no-op, not the base decoder/executor loop.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
NSR that scales [14]	ICML / 2021	Set Transformer encoder and Transformer decoder	Prefix-notation equation skeletons; operators; constant placeholders	Beam search over skeletons plus BFGS constant fitting	Encode to decode skeletons to fit constants to select candidate	Task harness; neural equation proposal + symbolic numeric-fitting coupling	Artifact synthesis / induction	Level 1	Hybrid mainly through symbolic output space and post-decoding refinement.
NeuroComparatives [38]	Findings of NAACL / 2024	Generator LMs, contradiction classifier, and knowledge discriminator	Entity classes, prompts, lexical constraints, and the resulting comparative KB	Constrained decoding, contradiction filtering, and structured KB selection	Explicit retrieve to expand to generate to filter pipeline	Retrieval / curation workflow; generate-to-constraint-filter KB coupling	Artifact synthesis / induction	Level 2	Symbolic constraints matter, but reasoning remains relatively shallow.
Hierarchical rule induction [35]	ICML/ 2022	Predicate embeddings; soft-unification parameters	Meta-rules; proto-rules; predicates; auxiliary predicates	Layered forward chaining with soft unification	Fixed hierarchical layered schedule	Task harness; soft-unification rule-learning coupling	Artifact synthesis / induction	Level 4	Theoretical expressivity analysis strengthens the symbolic interpretation.
PhyE2E [89]	Nature Machine Intelligence / 2025	OpenLLaMA2-3B generator; oracle NN; transformer formula model	Formulas; expression trees; physical units; grammar pool	D&C decomposition; MCTS; GP; constant optimization	Explicit multi-stage pipeline from generation to refinement	Scientific task harness; staged generate-decompose-refine coupling	Artifact synthesis / induction	Level 4	Controller-heavy reading is plausible, but the lower primary level is safer.
AlphaGeometry [79]	Nature / 2024	Transformer language model trained from scratch on synthetic theorem-proof data to propose auxiliary constructions under beam search.	Deductive database (DD), algebraic reasoning (AR), proof DAGs, traceback, proof pruning, and formal geometry premises/statements.	Symbolic deduction closure, algebraic deduction, proof search, traceback, and pruning guided by neural construction proposals.	Explicit alternating proof-search loop that interleaves symbolic deduction with language-model construction proposals under beam search.	Formal theorem goal; LM-guided proof-search coupling	Artifact synthesis / induction	Level 4	This is the strongest and most formal system in the source set, but its scope is limited to a specialized geometry environment.
SymbolNet [80]	Machine Learning: Science and Technology / 2025	Operator-structured symbolic-layer network with trainable weights and thresholds	Mathematical operators and extracted symbolic expressions	Minimal; pruning and expression extraction rather than explicit symbolic inference	Single-phase pruning/training framework with target sparsity ratios	Task harness; symbolic-expression layer / pruning coupling	Artifact synthesis / induction	Level 0	Symbolic output is central; symbolic reasoning is not.
Boundary and adjunct cases									

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Experimental NL reasoning pipeline [78]	CADE / 2023	Pretrained Stanza parser used to obtain Universal Dependencies graphs.	Extended first-order logic with confidences and blocker literals, object databases, and a large knowledge base feeding the GK reasoner.	Resolution-style automated reasoning with defaults, blockers, answer clauses, confidence handling, and proof/counterproof search.	Sequential pipeline orchestration by the driver plus internal blocker-checking and proof-search control inside GK.	User question + driver pipeline; parse-to-logic-to-reason coupling	Boundary / adjunct	Level 1	Best understood as a symbolic reasoning backbone with a neural parsing front-end and explicit plans for future hybridization.
bears [65]	UAI / 2024	Ensemble of concept extractors	Inherited knowledge/reasoner from backbone	Mostly inherited from base NeSy system	Sequential ensemble training; averaging	Task harness; ensemble concept extraction over inherited symbolic reasoner	Boundary / adjunct	Level 0	Strong calibration wrapper, weak standalone NeSy identity.
CONVFINQA [18]	EMNLP / 2022	Evaluated retriever-generators and prompting models	FinQA DSL programs	Program generation/execution in evaluated baselines; dataset decomposition pipeline	No-op runtime controller: dataset-construction/evaluation workflow sequences program generation and execution, but no deployed adaptive controller is introduced	Dataset / benchmark workflow; executable-program benchmark coupling	Boundary / adjunct	Level 0	Benchmark-centric entry; executable-program structure is present, but runtime control is no-op beyond the fixed evaluation workflow.
CORRPUS [28]	Findings of ACL / 2023	Codex; GPT-3/BART in parts	Python classes; world state; attributes; generated code	Prompted state tracking; contradiction detection	Fixed/no-op controller: prompt-template scaffold sequences generated code and world-state updates without adaptive control	Story prompt + benchmark tasking; prompt-scaffolded code/world-state coupling	Boundary / adjunct	Level 0	Symbolic world-state scaffolding matters, but controller behavior is fixed/no-op and strong symbolic reasoning is not fully specified.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Empirical Investigation of Neural Symbolic Reasoning Strategies [7]	Findings of ACL: EACL / 2023	T5-base; T5-large; BART-base	Equation strings and intermediate chain strings only	Neural generation of reasoning chains; no-op symbolic runtime over equation/intermediate chain strings; no external symbolic solver is invoked	Fixed/no-op strategy controller: reasoning strategy is fixed by experimental condition rather than adaptively selected	Benchmark harness; symbolic-text task with no-op runtime symbolic module	Boundary / adjunct	Level 0	Symbolic task format is present but the runtime symbolic module is no-op; keep Level 0 unless a real symbolic execution/checking component is foregrounded.
HMC error recovery [51]	CIKM / 2024	Base vision model; secondary model; optional binary models; downstream LTN learner	EDR rules; violating sets; recovered hierarchy constraints	Rule induction via submodular-ratio optimization; rule-based error detection	Fixed pipeline from predictions to rules; optional transfer into LTN	Benchmark pipeline; prediction-to-rule recovery with optional LTN transfer	Boundary / adjunct	Level 0	Core method is a checker/recovery layer; downstream constrained learning makes a higher reading plausible.
FSNS feature selection [36]	ACM TIST / 2025	RL collector; variational transformer	Feature-ID token sequences	Embedding-space search	Multi-stage pipeline	Offline optimization workflow; tokenized feature-search coupling	Boundary / adjunct	Level 0	Symbolic role is chiefly token-level and representational.
HLB VSA [5]	NeurIPS / 2024	Downstream neural systems using HLB	VSA binding/unbinding algebra	Vector composition and retrieval	No-op controller: benchmark harness applies a fixed vector-binding/retrieval workflow; no adaptive NSAI control is introduced	Model-builder / benchmark harness; representational vector-symbolic coupling	Boundary / adjunct	Level 0	Important representational primitive, but not a full neuro-symbolic reasoner; controller role is no-op in the evaluated workflow.
KLAY [63]	ICLR / 2024	External neural network outputs leaf probabilities; no new neural model is proposed	Arithmetic circuits; d-DNNF/SDD structures; layered circuit representation	Exact circuit evaluation in semiring/log semiring form	Layerization + sequential layer execution	Model-builder / benchmark harness; neural leaf-probability to circuit-runtime coupling	Boundary / adjunct	Level 2	Infrastructure paper; lower reading preferred over a fuller system interpretation.
KG-driven zero-shot QA [62]	AAAI / 2021	GPT-2 and RoBERTa language models	ATOMIC, CSKG/CWWV resources, templates, and distractor-sampling rules	Mostly offline question construction and distractor selection	Fixed offline controller; no-op runtime controller after KG-guided dataset construction	Offline data-construction workflow; KG-guided supervision coupling	Boundary / adjunct	Level 0	Symbolic role is offline and data-centric; runtime controller is no-op beyond the trained evaluation workflow.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
LARS-VSA [66]	arXiv / 2024	HD attention; CNN/VGG encoders	Hypervectors; binding/bundling	Representational VSA operations over learned encodings; no separate symbolic solver	Fixed active evaluation controller: benchmark workflow coordinates neural encoding, hypervector operations, and downstream evaluation	Benchmark harness; representational hypervector coupling	Boundary / adjunct	Level 2	Symbolicity is representational rather than classical formal reasoning; Level 2 is defensible only if the fixed evaluation controller is treated as active rather than no-op.
LogiCity [55]	NeurIPS / 2024	Neural rendering; evaluated NN/NeSy baselines	FOL concepts/rules; SMT-grounded actions	SMT rule inference; benchmarked reasoning pipelines	Explicit simulator loop	Benchmark scenario generator; simulator-grounded task coupling	Boundary / adjunct	Level 0	Benchmark artifact more than single deployed NS system.
MDD-5k [88]	AAAI / 2025	Doctor/patient/tool LLM agents	Dynamic diagnosis tree; knowledge graph; structured case fields	Topic planning; tree traversal; experience parsing	Strong tool-agent orchestration	Case schema + tool-agent workflow; controller-guided dialogue-synthesis coupling	Boundary / adjunct	Level 3	Explicit controller/tool-agent structure is real, but the paper is primarily dataset/framework oriented.
NeSy code-comment workflow [1]	GeNeSy @ ESWC / CEUR / 2024	ChatGPT 3.5 + downstream classifiers	17-rule ruleset; generated script	Rule-guided generation only; no-op runtime reasoner beyond fixed rule application	Fixed workflow controller; no-op runtime controller	User workflow prompt; rule-guided data-generation coupling	Boundary / adjunct	Level 0	Most convincing as a neuro-symbolic data-generation workflow; runtime reasoning/control roles are no-op beyond the fixed rule-guided workflow.
PANoS verification [50]	IJCAI / 2024	Observation/perception function implemented by a neural network	Agent protocols; transitions; interpreted-system states; CTL formulas	Formal abstraction; simulation arguments; model checking	Symbolic transition semantics over neural perceptions	Specification engineer; neural perception under symbolic transition / proof coupling	Boundary / adjunct	Level 1	Higher reading is possible if the verifier is foregrounded; lower level kept for conservatism.
RETOMATON [6]	ICML / 2022	Base LM; hidden-state encoder; pLM distribution	Weighted finite automaton; clustered states; pointers; datastore entries	Automaton traversal and transition-weighted retrieval approximation	Explicit restart-or-traverse policy using threshold tau	LM inference harness; restart-or-traverse automaton coupling	Boundary / adjunct	Level 3	Symbolic automaton is explicit, but mainly retrieval-oriented.

Continued on next page

Table A1 continued from previous page

Paper	Venue / Year	Neural / Sub-Symbolic	Symbolic Store	Reasoning Process	Neuro-Symbolic Controller	Orchestrator / Coupling	Architecture family	Neuro-Symbolic Level	Notes
Ontology-embedding PPI benchmark notebooks [53]	Briefings in Bioinformatics / 2021	Ontology embeddings and learned similarity/prediction models, including Onto2Vec, OPA2Vec, EL Embeddings, TransE, random-walk/Node2Vec embeddings, and SiameseNN variants	Gene Ontology; OWL axioms; GO annotations; protein/function/location associations; ontology-derived graphs; PPI knowledge base	OWL entailment/deductive closure, ontology-to-graph conversion, semantic similarity, superclass/true-path propagation, and PPI prediction evaluation	Notebook-driven benchmark workflow coordinating data preparation, ontology/graph construction, embedding generation, similarity or supervised prediction, and evaluation	Researcher/notebook harness coupling symbolic GO axioms and ontology-derived representations to ML-based PPI prediction	Boundary / adjunct	Level 0	Reproducible ontology-ML benchmark workflow rather than survey-only evidence; conservative Level 0 because the workflow is primarily a benchmark demonstration rather than a deployed integrated architecture.
Travel demand with DT rules [2]	arXiv / 2025	Deep neural network	Decision-tree rules as binary features	Rule extraction and feature matching	Fixed offline preprocessing controller; no-op runtime controller	Offline model-building pipeline; rule-feature augmentation coupling with no-op runtime control	Boundary / adjunct	Level 0	Rules function mainly as engineered features; runtime symbolic control is no-op, so Level 0 is appropriate.
Vehicle [22]	FSCD / 2025	Neural controller declarations / trained networks	VCL specifications, properties, proofs, verifier queries, and symbolic program semantics	Proof decomposition, query generation, backend compilation, and formal integration	Explicit cross-backend compiler/toolchain coordinating training, verification, and ITP export	Specification engineer + verifier backends; compilation / proof-integration coupling	Boundary / adjunct	Level 6	Level mapping is imperfect because the paper is primarily verification infrastructure rather than a standard task-solving runtime system.